

AD-A101 319

L N K CORP SILVER SPRING MD  
KNOWLEDGE-BASED IMAGE ANALYSIS.(U)

APR 81 G C STOCKMAN, B A LAMBIRD, D LAVINE

ETL-025B

F/G 9/2

DAAK70-77-C-0110

NL

UNCLASSIFIED

1-3  
2-3



ETL-0258

ALA101319

Knowledge-based image analysis



George C. Stockman

Barbara A. Lambird

David Levine

Laveen N. Kanal

L.N.K. Corporation  
302 Notley Court  
Silver Spring, MD 20904

APRIL 1981

DMC FILE COPY

Approved for Public Release; Distribution unlimited



Prepared for  
U.S. ARMY CORPS OF ENGINEERS  
ENGINEER TOPOGRAPHIC LABORATORIES  
FORT BELVOIR, VIRGINIA 22060

81 7 13 078



---

Destroy this report when no longer needed.  
Do not return it to the originator.

---

The findings in this report are not to be construed as an official  
Department of the Army position unless so designated by other  
authorized documents.

---

The citation in this report of trade names of commercially available  
products does not constitute official endorsement or approval of the  
use of such products.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

cont

detected image structure.

By using iconic knowledge, the image interpretation paradigm becomes a three step process. First, some primitive features of the imagery must be recognized without any area-specific knowledge. Second, the imagery is aligned or registered with the knowledge base by drawing correspondences between the image features and their iconic analogues in the GDB. The matching is formalized by derivation of a transformation which maps points (x,y) of the image to points (u,v) in GDB coordinates. The final step of the process is the analysis of those parts of the image which were not successfully interpreted in steps 1 and 2. This implies a top-down search for image structures which correspond to features in the GDB.

Section 2 of the report treats primitive extraction. The emphasis is currently on lineal, point, and region features only. A method for automatically inferring a rotation and translation transforming image to map is given in Section 3. Classification of registered regions is discussed in Section 4. Verification of lineal GDB features in gray-scale imagery is introduced in Section 5.

Accession For	
NTIS	GRA&I <input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A	

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

## UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 19 ETL	2. GOVT ACCESSION NO. 19 0358 / AD-A102379	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 KNOWLEDGE-BASED IMAGE ANALYSIS		5. TYPE OF REPORT & PERIOD COVERED Contract Report 1 Sept 1978 - 1 Dec 1980
7. AUTHOR(s) 10 George C. Stockman, Barbara A. Lambird, David Lavine, and Laveen N. Kanal		8. CONTRACT OR GRANT NUMBER(s) 15 DAAK70-77-C-0110
9. PERFORMING ORGANIZATION NAME AND ADDRESS L.N.K. Corporation 302 Notley Court Silver Spring, MD 20904		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 12 244
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060		12. REPORT DATE 11 April 1981
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060		15. SECURITY CLASS. (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) registration, image matching, map-guided cartography, feature extraction, verification, region classification, pattern recognition, image analysis.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  The work reported was directed toward employing a priori knowledge in the automatic analysis of aerial imagery. Major objectives of the research were directed toward (1) map-guided registration, (2) verification of geographic data bases extracted from imagery, (3) enrichment of geographic data bases, and (4) automatic terrain feature extraction using multiple sources of knowledge and multi-level decision making. The key component in all of the work was the matching of existing iconic structure in a geographic data base (GDB) with		

Table of Contents  
Knowledge Based Image Analysis

Preface . . . . .	1
1. Introduction . . . . .	1
1.1 Problems addressed. . . . .	1
1.2 L.N.K. related research and software development . . . . .	3
1.3 Sample data sets . . . . .	3
1.4 Laboratory configuration, parameters, and definitions . . . . .	9
1.5 Image processing conventions and methodology . . . . .	12
2. Feature Detection . . . . .	13
2.1 Gradient operator and masking . . . . .	14
2.2 Continuous edge linking . . . . .	20
2.3 Hough detection of straight edge elements . . . . .	23
2.3.1 Experiments with primitive detection of edge elements . . . . .	27
2.4 Using ROSA frequency domain as edge detector . . . . .	32
2.5 Edge extension . . . . .	35
2.6 Intersection detection . . . . .	36
2.6.1 Experiments with edge extension and intersection detection. . . . .	39
2.7 Intersection classification . . . . .	44
2.7.1 Merging line segments . . . . .	44
2.7.2 Pairing line segments . . . . .	47
2.7.3 Detecting and classifying intersections . . . . .	49
2.7.4 Experiments on detection and classification of intersections	50
2.8 Region segmentation . . . . .	60
3. Registration of image data to a map . . . . .	62
3.1 Survey of previous registration work . . . . .	63
3.2 L.N.K. registration procedure . . . . .	66
3.2.1 The general procedure . . . . .	66
3.2.2 Specific registration procedure . . . . .	68
3.2.3 A simple sample . . . . .	68
3.2.4 A more complex example . . . . .	71
3.2.5 Details of the L.N.K. registration procedure . . . . .	77
3.2.5.1 Pairing edge elements and populating $\alpha$ -space . . . . .	78
3.2.5.2 Clustering in $\alpha$ -space . . . . .	81
3.2.5.3 Evaluating the goodness of fit for $\alpha$ . . . . .	84
3.3 Registration experiments . . . . .	87
3.3.1 4621 image . . . . .	87
3.3.1.1 Registration using real edges . . . . .	92
3.3.1.2 Robustness of registration procedure . . . . .	92
3.3.2 Registration of a stereo pair . . . . .	95
3.3.3 Conclusions on 2-D Registration . . . . .	101
3.4 Full RST transformation . . . . .	103
3.4.1 Example with scale . . . . .	108
3.5 Extending LNK registration to 3 dimensions . . . . .	112

4. Region Identification . . . . .	122
4.1 Region label weights . . . . .	123
4.1.1 Disjoint principal components model . . . . .	125
4.1.2 Pooled covariance discrimination . . . . .	129
4.1.3 Synthetic data experiments . . . . .	131
4.1.4 Discussion of region weight evaluation . . . . .	138
4.1.5 Discussion of region label rankings . . . . .	139
4.2 Overview of region matching . . . . .	142
4.2.1 Region image matching using similarity of region features .	143
4.2.2 Graph theoretical region matching . . . . .	145
4.2.2.1 Region Matching Experiments . . . . .	151
4.2.2.2 Graph Matching Discussion . . . . .	159
5. Verification of Linear Features . . . . .	160
5.1 Techniques used in verification process . . . . .	161
5.1.1 Profile search . . . . .	161
5.1.2 Quality Evaluation . . . . .	162
5.1.3 Servoing . . . . .	164
5.2 Examples from 4621 . . . . .	166
5.3 Examples from DREUX 13 . . . . .	171
5.4 Discussion and conclusions on verification . . . . .	182
6. Summary of Conclusions . . . . .	185
7. References . . . . .	188
Appendix A. Software Overview . . . . .	190
A.1 Research Image Processing System (RIPS) . . . . .	190
A.2 Edge Extraction Routines EDGEX and EDGEY . . . . .	191
A.3 Registration Software REG . . . . .	192
A.4 Verification Software VERIFI . . . . .	193
Appendix B RIPS Software . . . . .	194
Appendix C Edge Extraction Routines EDGEX and EDGEY . . . . .	202
C.1 Loader commands for creating EDGEX . . . . .	204
C.2 Sample run of EDGEX . . . . .	205
C.3 Loader commands for creating EDGEY . . . . .	209
C.4 Sample run of EDGEY . . . . .	210
Appendix D Registration Software . . . . .	212
D.1 Procs containing common blocks used in registration . . . . .	216
D.2 Possible commands . . . . .	218
Appendix E Verification Software . . . . .	220
Appendix F Intersection Classification . . . . .	225

Preface

The work presented in this report was performed by L.N.K. Corporation scientists George C. Stockman, Barbara A. Lambird, David Lavine, and Laveen N. Kanal under contract DAAK70-77-C-0110 for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia 22060. Dr. Robert D. Leighty, Research Institute, USAETL served as contract technical monitor. The authors are indebted to Dr. Leighty and Mr. George Lukes, Research Institute, USAETL for helpful interaction and technical reviews during the performance of this work.

## 1. Introduction

This report describes research performed by L.N.K Corporation for USAETL during the approximate period of 1 Sept 1978 through 1 Dec 1980. The study was directed toward the use of knowledge in automatic image analysis and toward map-guided image analysis in particular and represents a continuation of studies for Wright Patterson Air Force Base by L.N.K. During the period of this report L.N.K. converted algorithms to run on a Hewlett-Packard real-time system at the Research Institute, USAETL, and experimented with these algorithms on imagery of interest to USAETL. The procedures and results are discussed in Sections 2 to 6.

### 1.1 Problems addressed

Past research has indicated that a large amount of prior knowledge must be brought to bear on the problem of interpreting imagery whether it is done by man or machine. Image analysis by computer thus leads into studies of knowledge representation and application, which is currently a very active field in artificial intelligence (A.I.). The work reported here limits the form of possible knowledge sources to those which relate to the 2-D or 3-D spatial geometry of the real world. The resulting knowledge is "iconic" and can be represented in a fashion similar to the geographic data base (GDB) that contains encodings of the earth's features addressed by geographic coordinates.

By using iconic knowledge, the image interpretation paradigm becomes a three step process. First, some primitive features of the imagery must be recognized without any area-specific knowledge. Typically

these features would be major lineals or boundaries of objects such as ponds or buildings. Secondly, the imagery is aligned or registered with the GDB by drawing correspondences between the image features and their iconic analogues in the GDB. The matching is formalized by derivation of a transformation which maps points (x,y) of the image to points (u,v) in GDB coordinates. The third and final step in the paradigm is to analyze the remaining parts of the image which were not successfully interpreted in steps 1 and 2. This implies a top-down search for image structures which correspond to features in the GDB.

The paradigm is not the most general. It assumes that a map or GDB already exists. Also it does not provide for all forms of knowledge -- for instance, that roads tend to intersect or that water runs downhill. However, there are important problems to study. For step 1 we need to find feature extraction procedures that are reliable enough to detect major features in a variety of imagery. For step 2 we need to have a method of determining a global registration transform using the ambiguous and errorful information from local feature correspondences. Finally, for step 3 a method of verifying GDB features in the imagery is required as well as a method for recording positive change which should be entered into the GDB. These three problems are addressed in Sections 2,3, and 4 respectively. Section 2 treats primitive extraction. The emphasis is currently on lineal, point and region features only. A method for automatically inferring a rotation and translation transforming image to map is given in Section 3. Classification of registered regions is discussed in Section 4. Verification of lineal GDB features in grey-scale imagery is introduced in Section 5.

## 1.2 L.N.K. related research and software development

More than 2 years of related research by L.N.K. for Wright Patterson Air Force Base has been reported in Stockman [1979]. The techniques and software developed in that study were carried forward into the current research. Thus results reported here represent a second iteration of improvement of techniques and testing on a wider variety of images. Prior work was done on digitized images stored on disk files. This report describes image processing performed using on-line analogue image storage sampled by a 32x32 pixel digital array sensor positioned under program control. The three software subsystems implemented at USAETL are the feature detection subsystem discussed in Section 2, the registration subsystem discussed in Section 3, and the verification subsystem documented in Section 5. These subsystems are detailed in the Appendices.

## 1.3 Sample data sets

The experiments discussed in Sections 2 thru 5 used the images shown in Figures 1.1 thru 1.3. Tables 1.1 and 1.2 give coordinates of selected points labeled on the images. This data was chosen because of rich straight line structure which the existing techniques had been developed to handle. Terrain with less man-made structure will be handled later when feature extraction techniques are further developed.

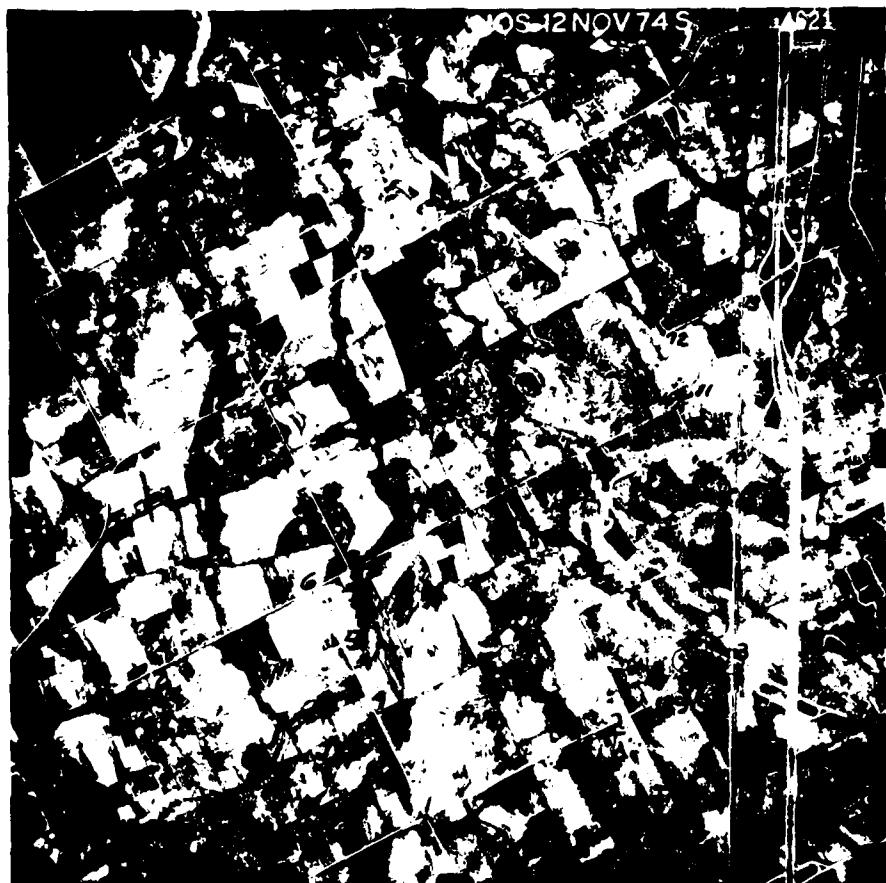


Figure i.1 Aerial photo over midwestern U.S. referred to as "4621"

Table 1.1      Digitization of points on 4621 image using Talos digitizer,  
 (0.001 inch resolution) and scanning stage (0.0005 inch  
 resolution)

IMAGE PT #	TALOS		STAGE		DESCRIPTION
	X	Y	X	Y	
1	2498	3641	5000	5000	X (origins)
2	4365	2747	8722	6803	T
3	5583	2191	11120	8052	X
4	3044	4744	3913	7203	T
5	3342	5357	3314	8405	T
6	3611	5908	2814	9511	T
7	1980	6678	-	-	T
8	3975	5750	3508	9946	T
9	5516	5073	6563	11448	T
10	7130	3514	10964	12108	T
11	7430	4091	10367	13304	L
12	7723	4700	9866	14600	L
13	8642	4258	-	-	X
14	6952	3223	11209	11461	A
15	6942	3613	10565	12000	L
16	7233	4181	10010	13101	T
17	4861	7575	1710	13600	X
18	5980	7667	-	-	Y
19	6444	7480	-	-	T
20	9471	6190	-	-	T
21	5013	7886	1399	14295	A
22	4480	7729	-	-	T
23	7346	9632	-	-	A
24	9282	5763	-	-	A
25	9700	6104	-	-	Y
26	5279	7999	1513	14804	A
27	4010	1991	9388	5480	A
28	5480	2552	-	-	L
29	4808	2527	-	-	A



Figure 1.2 Aerial photo over France referred to as "DREUX 13". (See Figure 1.3 for stereo mate.)

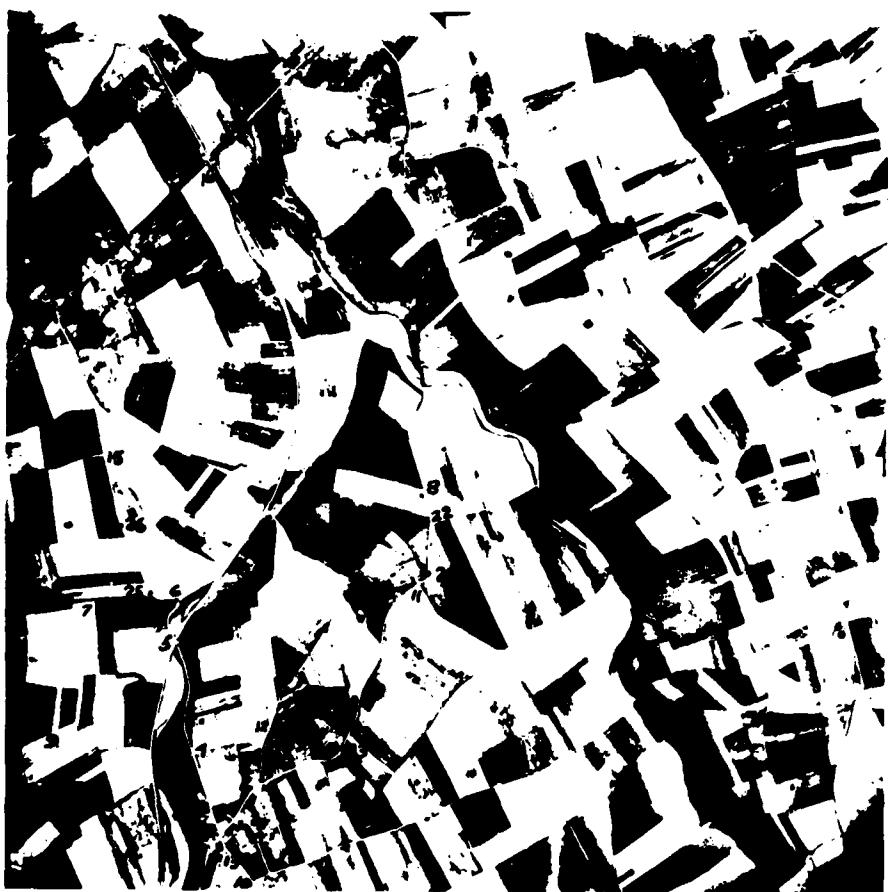


Figure 1.3 Aerial photo over France referred to as "DREUX 12" (See Figure 1.2 for stereo mate).

Table 1.2      Digitization of points on Dreux image (using Talos digitizer,  
0.001 inch resolution.)

IMAGE PT #	DREUX 13		DREUX 12		DESCRIPTION
	X	Y	X	Y	
1	8141	3564	7004	2264	field corner (origin)
2	7199	3140	6271	1567	Y road intersect
3	7016	2615	6277	1014	Y road intersect
4	6514	3255	5573	1448	T road intersect
5	5954	4274	4723	2247	X intersect, road & RR
6	5858	4654	4504	2563	nose point on road
7	5073	4382	3846	2055	corner of dark field
8	7858	6572	5756	4985	corner of dark field
9	5070	2974	4307	733	corner of dark field
10	7493	2257	6835	823	T road intersect
11	8128	5648	6313	4207	X or T road intersect
12	7160	3723	6041	2103	T road intersect
13	7115	2473	6414	909	Y road intersect
14	7621	2390	6921	982	T road intersect
15	4680	5822	3007	3277	T road intersect
16	4676	9032	1973	6288	X intersect, road & RR
17	3981	7597	1788	4724	T road intersect
18	3719	7055	1726	4125	X road intersect
19	3559	6595	1705	3644	Y road intersect
20	2940	4797	-	-	Y road intersect
21	2417	3261	-	-	T road intersect
22	8052	6252	6055	4737	Y road intersect
23	7910	3819	6720	2432	T road intersect
24	8178	5769	6342	4313	on straight of road
25	5551	4671	4226	2475	on straight of road
26	5062	5221	3606	2802	on straight of road

#### 1.4 Laboratory configuration, parameters, and definitions

The laboratory environment used for the experiments is described here and is diagrammed in Figure 1.6. As Figure 1.6 shows, the ROSA system allows a computer program to directly interact with an image stored on film. A small window on the film (about 1/8") can be imaged onto a 32 by 32 element photo diode array which is then sampled and converted to 10-bit digital values for consumption by the program. The end result is 1024 integer values in a 2-D FORTRAN array. Using executive calls a FORTRAN program can position the stage to any point in the selected 5 inch x 5 inch area and on a CRT display the user can view the current window.

Using a mirror as a switch, the laser beam passing through the film could be sent to an electro-optical subsystem which computes the Fourier Transform and makes it available to the program. The term "ROSA" originally meant "Recording Optical Spectrum Analyzer" and is still used although not descriptive of the current optical/digital sampling system which has evolved.

Table 1.3 contains a summary of the parameters and definitions used in the image processing environment. Note the difference in resolution of the different units of hardware. The stage used to move the film is driven in units of one-half mil. The photo-diodes of the array are, however, on 4 mil centers. The x-y-digitizer, on the other hand, has a resolution of 1 mil. Thus there are many scale changes required by a program which measures objects on the film and relates them to objects digitized by hand from a paper map.

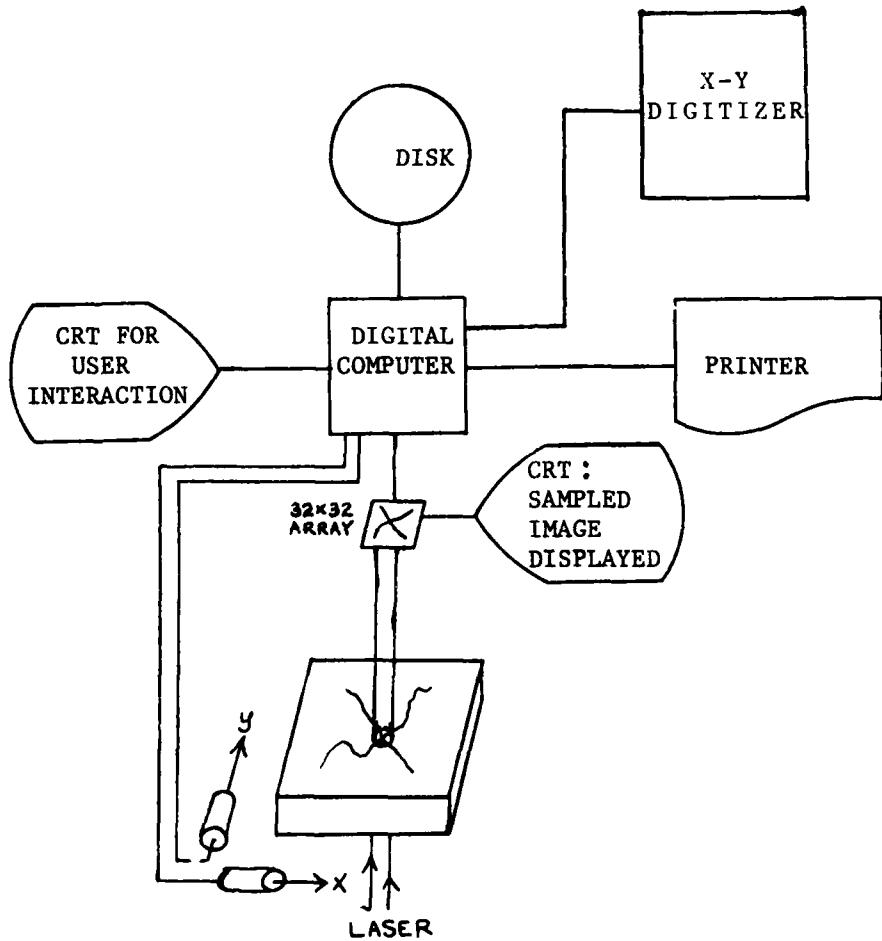


Figure 1.6 Diagram of ETL ROSA real time image sampling system.

Table 1.3

Parameters and definitions for ETL ROSA Lab environment

mil	one thousandth of an inch
pixel	a 10-bit integer measuring the intensity transmitted by a 4 mil x 4 mil square portion of film.
array	a 32 x 32 2-D array of light sensitive detectors used to sample film and create a 32 x 32 element 2-D matrix of pixels for computer processing.
stagel	stage element or resolution of stage movement which is 0.5 mil. Each pixel is thus 8 stagels on a side.
window	that portion of the film that can be sampled at a given stage position; representing an area 0.128 x 0.128 inches square or 32 x 32 pixels each 4 mils on a side.
registration	obtaining a transformation $T(x,y) = (u,v)$ which transforms an image point $(x,y)$ into its corresponding map point $(u,v)$ . In this report $T$ consists of a rotation $\theta$ with $0^\circ \leq \theta < 360^\circ$ and a translation no more than half the image diameter.

1.5 Image processing conventions and methodology.

Images placed on the stage were always addressed in stage coordinates with the "center" pixel,  $x=16$  and  $y=16$ , denoting the window position. An outstanding pass point was selected as a logical origin for each picture and computer processing was always begun with this point in the center of the window. The results reported here were often obtained by making several different runs with possible removal of the image from the stage in the interim. For the image 4621 the point #1 was chosen as the logical origin and was assigned coordinates  $x=5000$ ,  $y=5000$ , as shown in Table 1.1. Clearly, repeatability of the image addressing is dependent on the care taken to mechanically set up the film and viewing equipment (stage, mount, mirror, etc) and is probably no better than about 0.01 inch for the variety of experiments reported.

## 2. Feature Detection

Most L.N.K. studies have been concerned with lineal and point features only; regional analysis is addressed somewhat later in the section. By lineal features we mean continuous curves which define the boundary between regions of tonal contrast. Usually these curves define a feature of interest to the mapping community such as a stream, road, or side of a building. Field boundaries can also yield prominent lineals in an image and may or may not be of interest in the interpretation of imagery.

The beginning parts of this section of the report are devoted to edge detection. Here, the concept of lineal is almost synonymous with the term "edge" used in the literature. The fine difference is that a true lineal feature (i.e. a road or river) is 2-D and actually is characterized by 2 edges, one on each side of the road for instance.

Point features are defined from certain lineal features. For example, points of intersection or of high curvature can provide unique features of the image for registration and/or interpretation. Point features can also be defined as the centers of small identifiable regions such as buildings or ponds.

By using only lineals and points a very efficient image abstraction is rendered which can be of immediate use in registering images or in the partial interpretation of imagery. Some very successful registration experiments using only edge or point data are reported in Section 3. The rest of this section treats edge, point, and region extraction.

## 2.1 Gradient Operator and Masking

A point can be discovered to be an edge point by testing the tonal values in halfplanes on either side of the point. Figure 2.1 shows the 32 directions for edges that were used in the reported research. A general purpose routine (RPSML) exists which can differentiate either 8, 16 or 32 directions around the circle. Three "masks" for computing directional gradient values are shown in Figure 2.2. Direction  $d = 1$  is the vertical direction with higher tones at the right while direction  $d = 7$  is nearly horizontal with higher tones below. Given a pixel  $(x,y)$  in the image, the gradient magnitude in each direction  $d = 1, 2, \dots, 32$  can be computed by adding and subtracting tonal values as indicated by the masks in Figure 2.2. Division by a normalization factor is performed to take into consideration the number of pixels used and to make the magnitude geometrically isotropic. The magnitude and direction of the gradient at  $(x,y)$  is taken to be the magnitude and direction where a maximum is achieved. Note that for  $d = 17$  the mask for  $d = 1$  could be used with reversal of the sign on the magnitude, so that only 16 masks are actually applied. Only 4 directions are tried at resolution 8. The points selected for the masks sometimes differ from the ideal due to implementation considerations.

Points of weak gradient magnitude may be detected from a representation of the image as shown in Figure 2.3 which shows the results of a gradient computation on an image of an airplane wing. The arrows indicate gradient direction and show that the background tones are of higher value than the tones on the wing itself. The lower edge of the wing has orientation of about 13 while those on the top edge are roughly 30. Note that the "USAF" symbols on the wing create much structured gradient activity but that resolution is not fine enough for recognition.

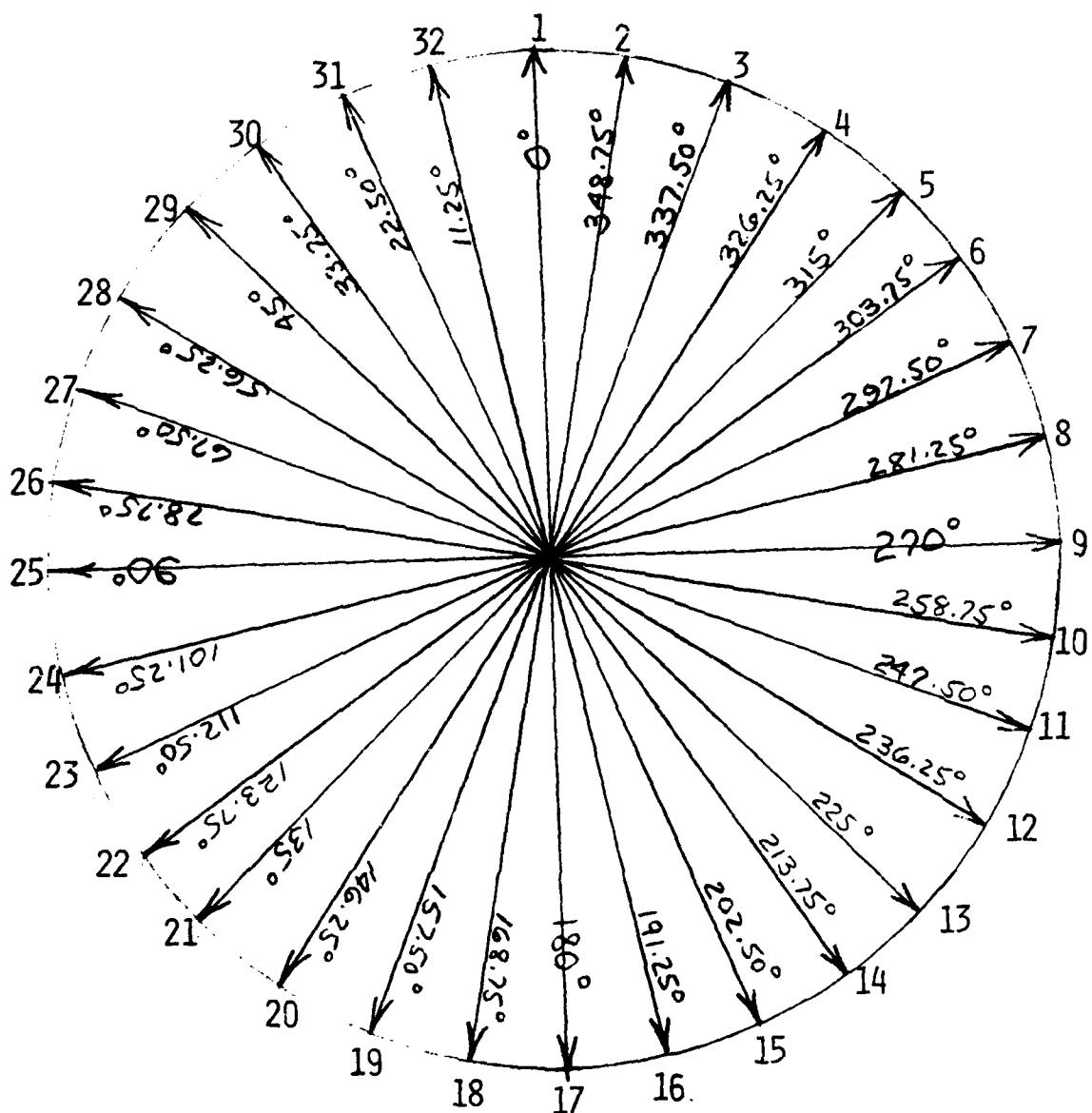


Figure 2.1 32-directional codes and associated angular heading in degrees.

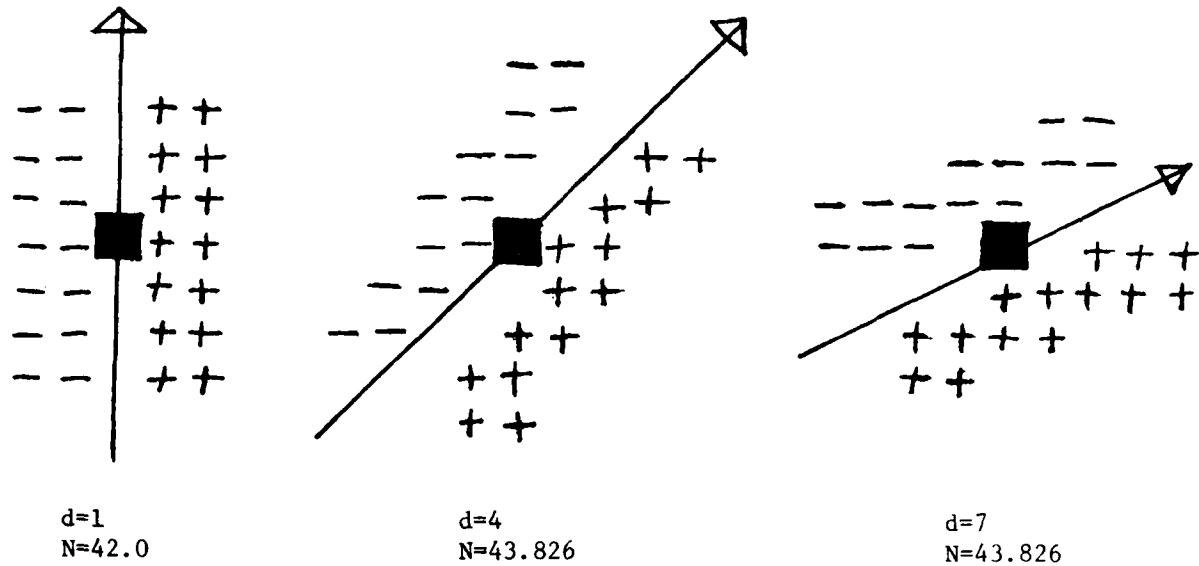


Figure 2.2 Three masks used for computing the gradient at a point in directions 1, 4, and 7. The directional resolution is one 32nd of the circle;  $N$  is the normalization factor. Edge directions of 1, 4, and 7 are indicated by the arrows.

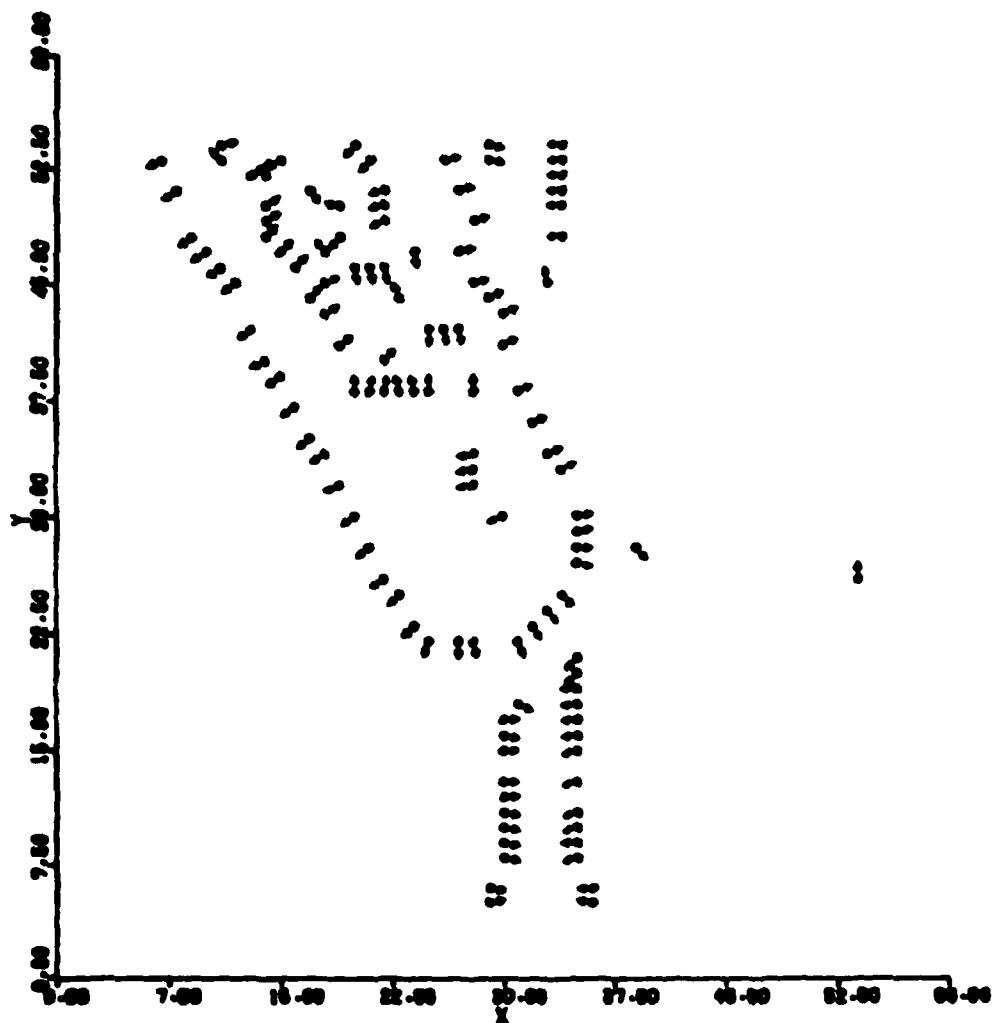


Figure 2.3 Gradient direction of high contrast points of right airplane wing.

Quite often, the presence of one edge will interfere with the detection of a neighboring edge. This is particularly true if only a limited fraction of the image points with high gradient magnitudes are selected to represent the image. In order to search for edges in the neighborhood of previously detected edges we could suppress the known edge points to remove the interference with the detection of other edge points. Conversely, to extend an existing edge segment we could suppress points of incompatible gradient directions in order to enhance the detection of continuing edge points.

Point masking can be done when the gradient image is computed by applying either locational or directional constraints (masks) as shown in Figure 2.4. A program EDGEY (see Appendix C) uses such masking - first to extend existing edges by suppression of competing gradient directions and then to suppress the existing edges in order to detect intersecting edges. In fact, to detect road intersections, two directional masks must be set, one for each side of the known road.

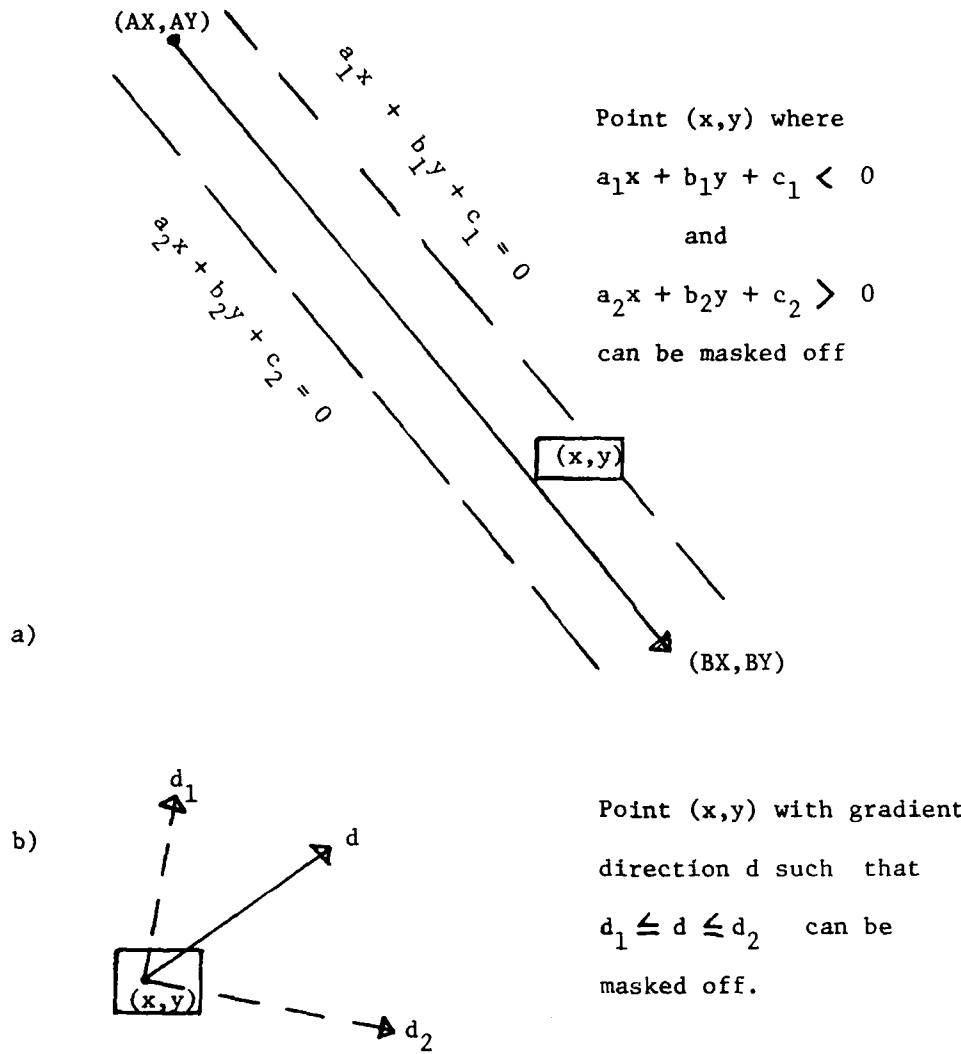


Figure 2.4

(a) Points near a known edge  $(AX, AY) - (BX, BY)$  can be masked off by their location with respect to two halfplanes or

(b) points with gradient direction in a certain range can be masked off.

## 2.2 Continuous Edge Linking

By examining the neighbors of edge pixels it is sometimes possible to determine the two pixels which continue the edge in the forward and backward direction. (Forward edge traversal by definition keeps higher tonal values toward the right.) Continuation can be determined from the spatial orientation of the neighbors and the gradient direction of the neighbors, all relative to that of the original pixel.

When determining which neighbor is the best forward and backward continuation a variable sized neighborhood is scanned in a spiral pattern for the first satisfactory neighbor. See Appendix B (/RPSCH/ common) for definition of the spiral pattern.

Stockman [1979] contains a detailed discussion of continuous edge linking. Figure 2.5 shows best forward and backward linking done for all pixels shown in Figure 2.3. (Done by program RPSWK.) A second global process can extract long chains of points which mutually link to each other and which have above threshold length. Figure 2.6 shows the long chains derived from the data in Figure 2.5. (Program RPSLK extracts the chains.)

Chains of 15 or more pixels were extracted from the imagery sampled in the ETL ROSA lab. Often these chains formed a straight edge which was usually also detected by the Hough detector (Section 2.3). Since interesting curved edges were not being formed due to the small 32x32 window size, the results of curve extraction were not used for further processing. In future experiments the effective window size could be enlarged and more interesting curved edges should be obtained.

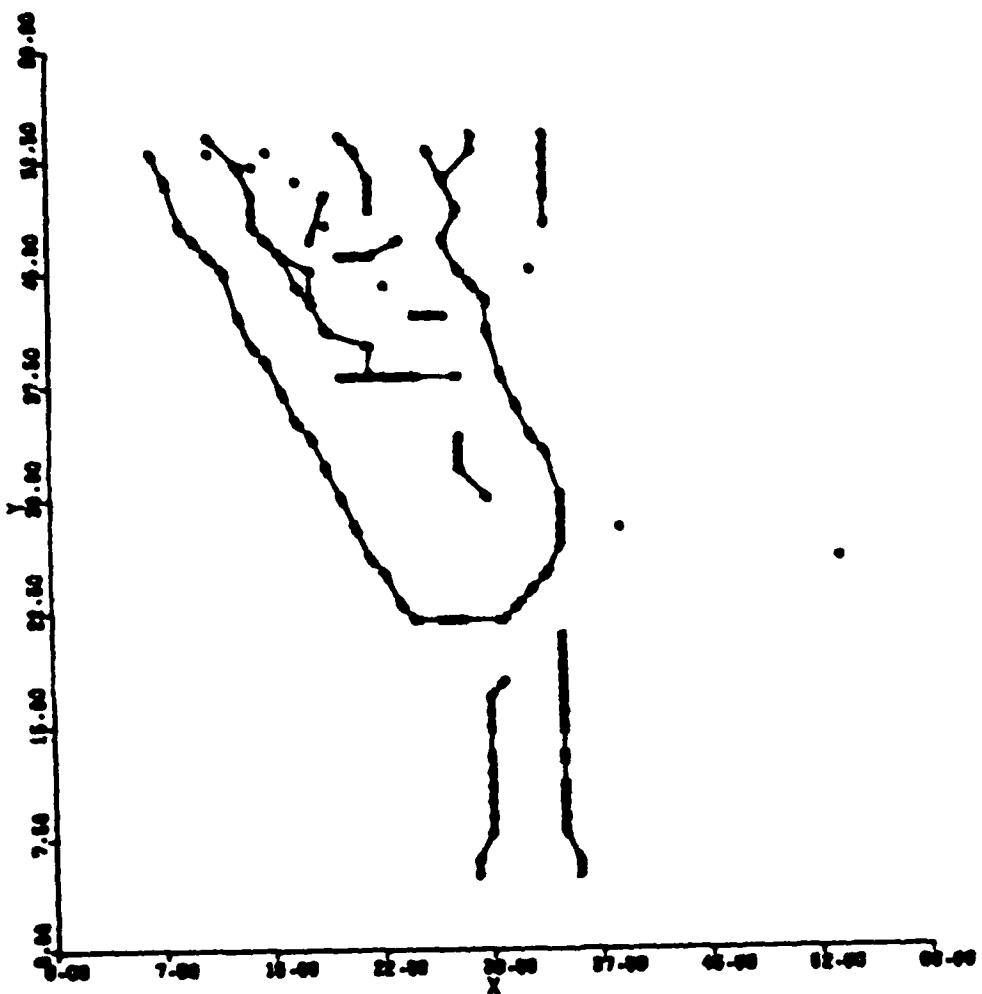


Figure 2.5 Plot of all forward and backward linking relationships among high contrast points of Figure 2.3.

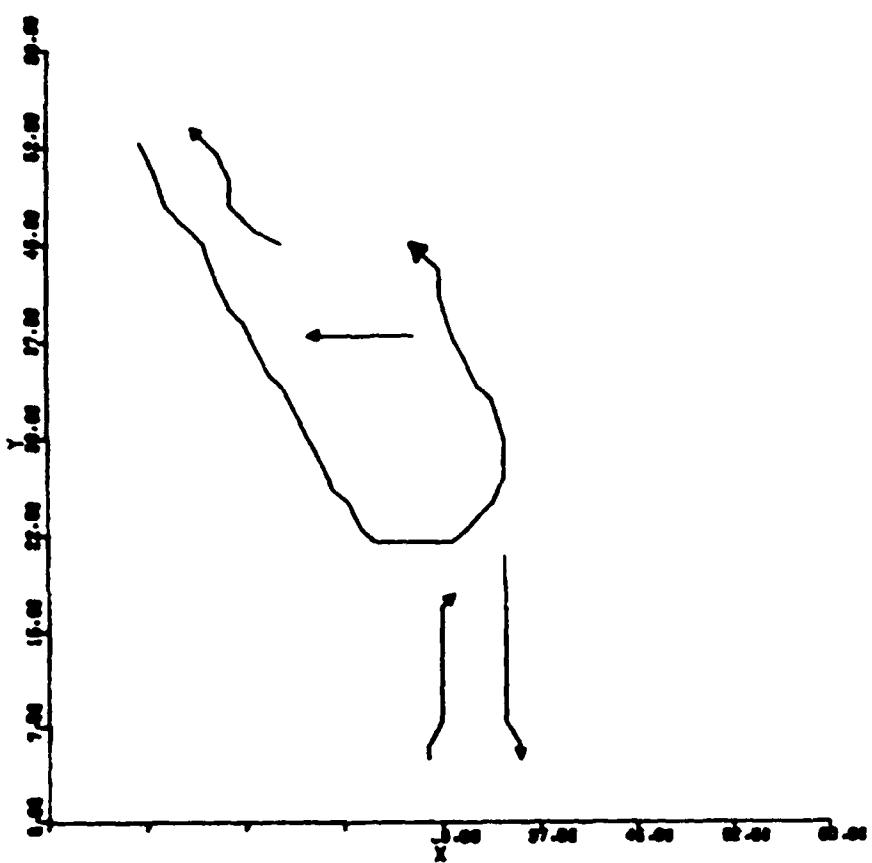


Figure 2.6 Long curve segments extracted from related points of Figure 2.5.

### 2.3 Hough detection of straight edge elements

The Hough transform is a popular device for the detection of linearity in a set of points. The mathematical development and practical use of the Hough transform is discussed in Duda [1972] and Stockman [1977]. The Hough detector was used heavily to achieve the results reported here and proved to be very reliable. The strategy used was to set stringent thresholds so that the false alarm rate would be nearly zero. In fact, no false alarms have been documented although not all of the thousands of windows processed have been studied. There were some "unwanted" detections on faint image structure such as field mowings and buildings which were difficult to discern on the CRT.

The windows sampled were 32x32 pixels, of which only the center 24x24 pixels were useable to the resolution 32 gradient operator, so detection thresholds were set at either 15 or 20 meaning that 15 or 20 pixels had to line up before a detection was signaled. Figure 2.7 documents the detection of a linear edge segment between a road and a field. Gradient directions 21 to 25 have been masked out so that one side of the road is nearly suppressed -- the side barely survives due to jitter in the gradient directions. The gradient magnitude histogram is shown which was used to select 38 points to represent the image window. The 38 points are all visible in the grey shade plot of the gradient image. The coarse accumulator array is defined for the 32 angular directions and radius values from -12 to +12: thus there are  $32 \times 25 = 800$  accumulators. Each of the selected 38 points can indicate an incrementing of those few accumulators defined by an R and  $\theta$  compatible with the gradient direction of the point. The entire set of resulting

TRIPLES ARE 0 .000E+00 .000E+00 .000E+00 ← locational masking not used  
 PAIRS ARE 1 21 25 ← mask off gradient directions between 21 & 25  
 GRADDL ON IMAGE=462L WINDOW= 5 28 5 28  
 GRADDL: NOW THIN (MAXHAG, TYPE= 96 0  
 HISTOGRAM OF GR 96 LEVELS, DIV= 2  
 0 1 2 3 4 5 6 7 8 9  
 456 6 31 9 14 7 3 1 4 2  
 5 1 3 0 3 2 1 1 3 0  
 2 0 2 3 0 0 2 2 0 2  
 1 1 0 0 0 2 0 0 1 1  
 8 1 1 0 2 0 0 1  
 MEAN=.219E+01 VARIANCE=.463E+02 NUMPIX= 576  
 SELECT: THRESHOLD= 22  
 SELECT: LENWRK= 38  
 \*\*\*VIRTUAL\*\*\* 6871 6902 5899 5930#SCREEN# 1 32 1 32 \*\*\*END\*\*\*  
 1234567890123456789012345678901234567890

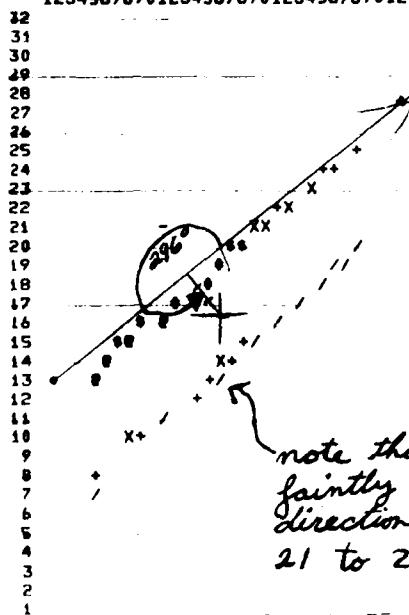


Figure 2.7 Point selection and Hough detection of straight edge along road.

accumulators is pictured in Figure 2.8. One side of the road is detected by peak detection in the accumulators at  $R = -3$  and  $\theta = 292^\circ$ .  $\theta = 292^\circ$  because that is the true geometric direction of tonal increase across the edge and  $R = -3$  because the edge is actually located 3 units in the direction  $292 - 180$  from the origin at pixel (16,16). The Hough detection is focused by contributing the 38 points to a refined accumulator array with only  $5 \times 7$  accumulators. In this process the edge location is refined to  $R = -3$  and  $\theta = 292^\circ$ . The peak response is diminished from 18 to 14 because the width of the edge "template" is now only 1 instead of 3 pixels. The detected edge element is reported to lie between points (1,13) and (32,28) in the image window and (6773,5897) and (7021,6017) in terms of global stage coordinates.

The focused accumulator uses  $2^\circ$  angular resolution but this resolution is not really obtainable with  $32 \times 32$  windows. The  $2^\circ$  unit was designed for the predecessor system which used  $64 \times 64$  windows where lines  $2^\circ$  apart differed by several pixels along their extent. In the research reported here, Hough detections were made in batch mode by sampling the imagery with a raster scan of butting windows. (Program EDGE3 is detailed in Appendix C.)

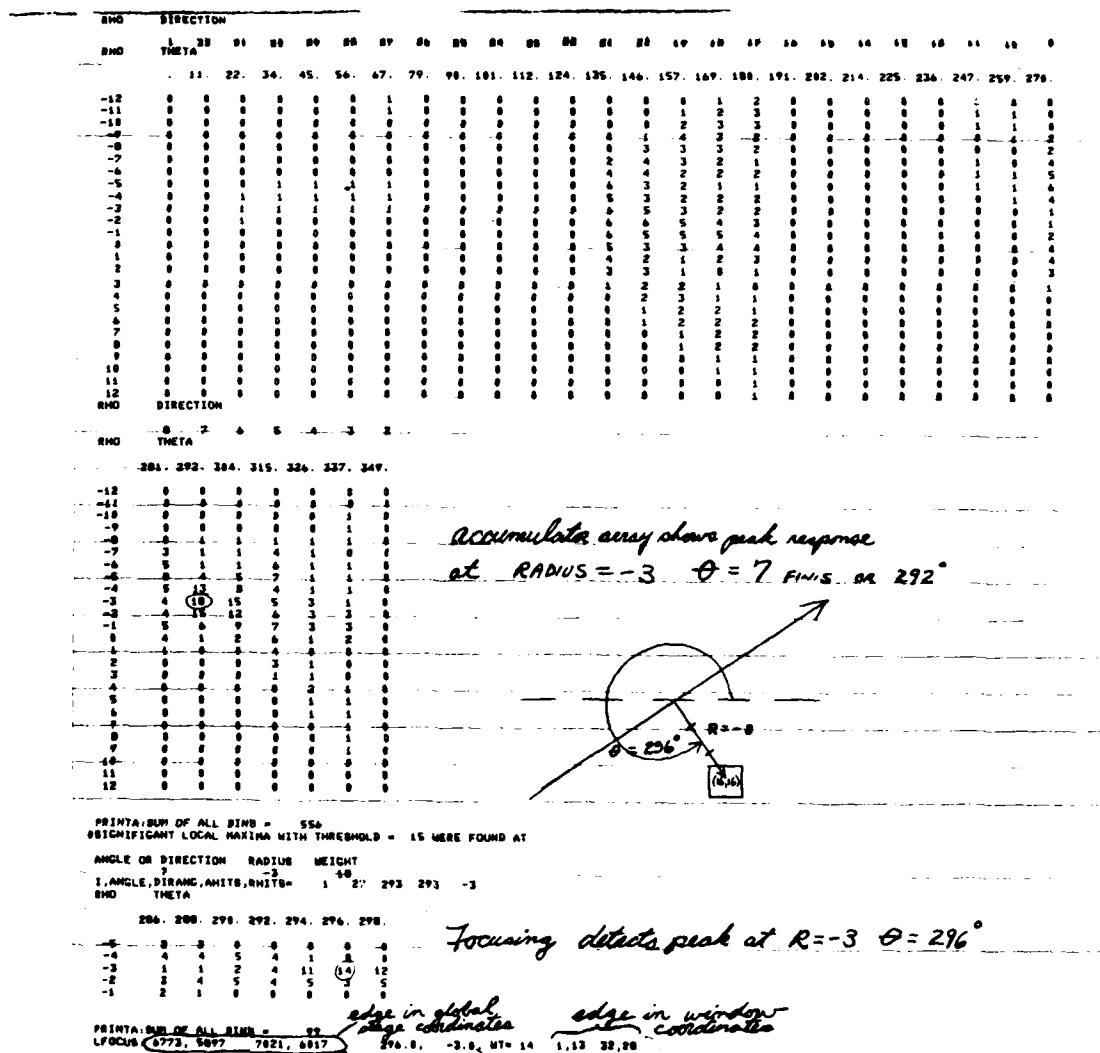


Figure 2.8 Coarse and fine accumulator arrays showing response to road edge shown in Figure 2.7.

### 2.3.1. Experiments with Primitive Detection of Edge Elements

The Hough detector was applied to over a thousand windows on the images 4621 and DREUX 13 (see Figures 1.1 and 1.2). Straight edge elements detected from 4621 are plotted in Figure 2.9 (a) while detections from DREUX 13 are plotted in Figure 2.9 (b). Parameters used for the detectors are given in Table 2.1. It should be noted that stringent thresholds were applied, especially on the number of points on a straight line so that few false alarms, if any, were reported. Only the straight edge elements are plotted; the curves extracted by the edge linking routine were ignored.

Many detections are evident along the major roads in 4621. The vertically appearing highway along the right edge of the image is well covered while the other roads in the image have only spotty coverage. Missed detections along the roads are attributed to one or more of the following effects:

- (1) unfavorable contrast due to sun angle and shadows,
- (2) gradient direction falls in between two of the 32 coarse level directions used by the Hough detector,
- (3) road cuts across a corner of the sampling window and not enough points exist to trigger detection,
- (4) road cuts through center of window and the 5% of the strongest gradient points are equally distributed on both sides of the road such that neither edge passes threshold.
- (5) in the neighborhood of other edges, such as other roads or field boundaries, the effect of (4) is compounded further.

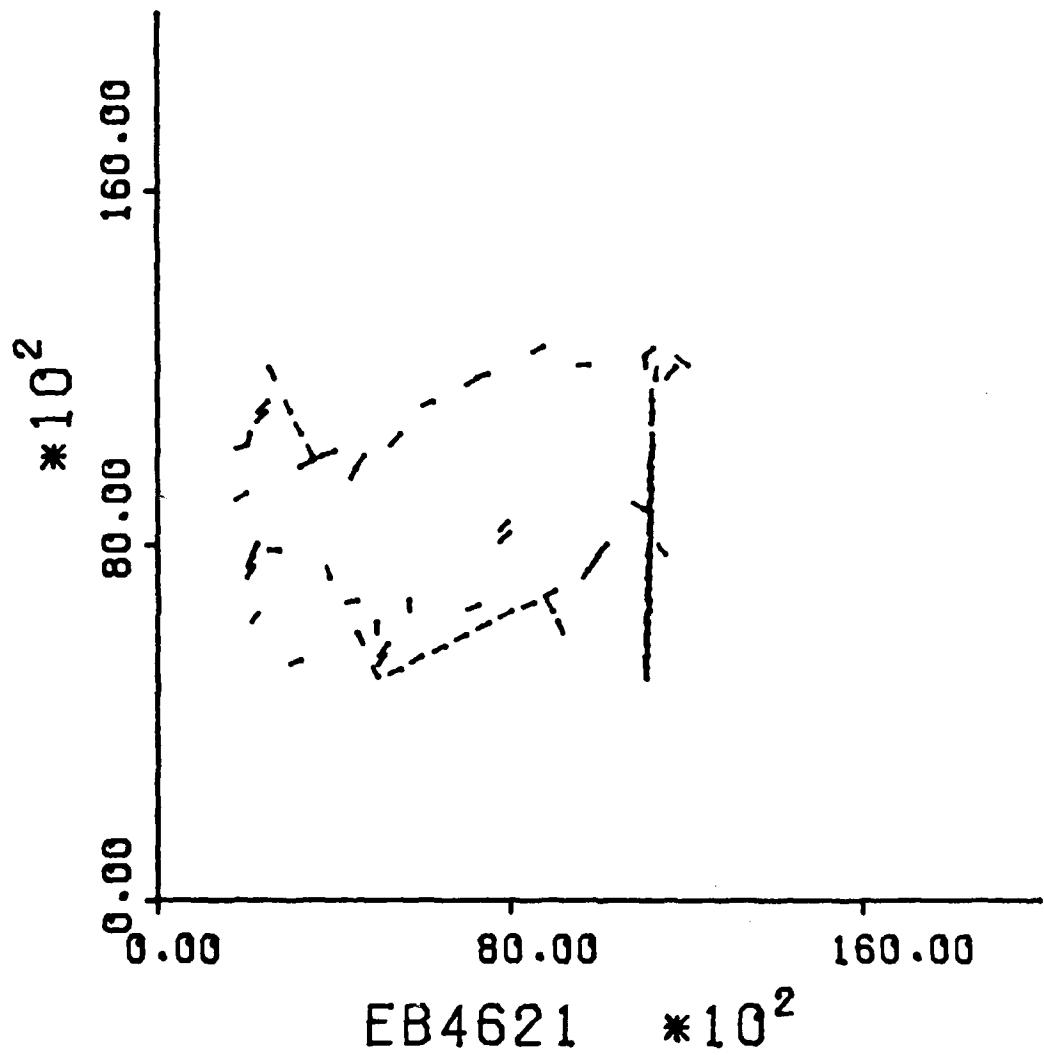


Figure 2.9(a) Plot of straight image edges extracted from 4621.

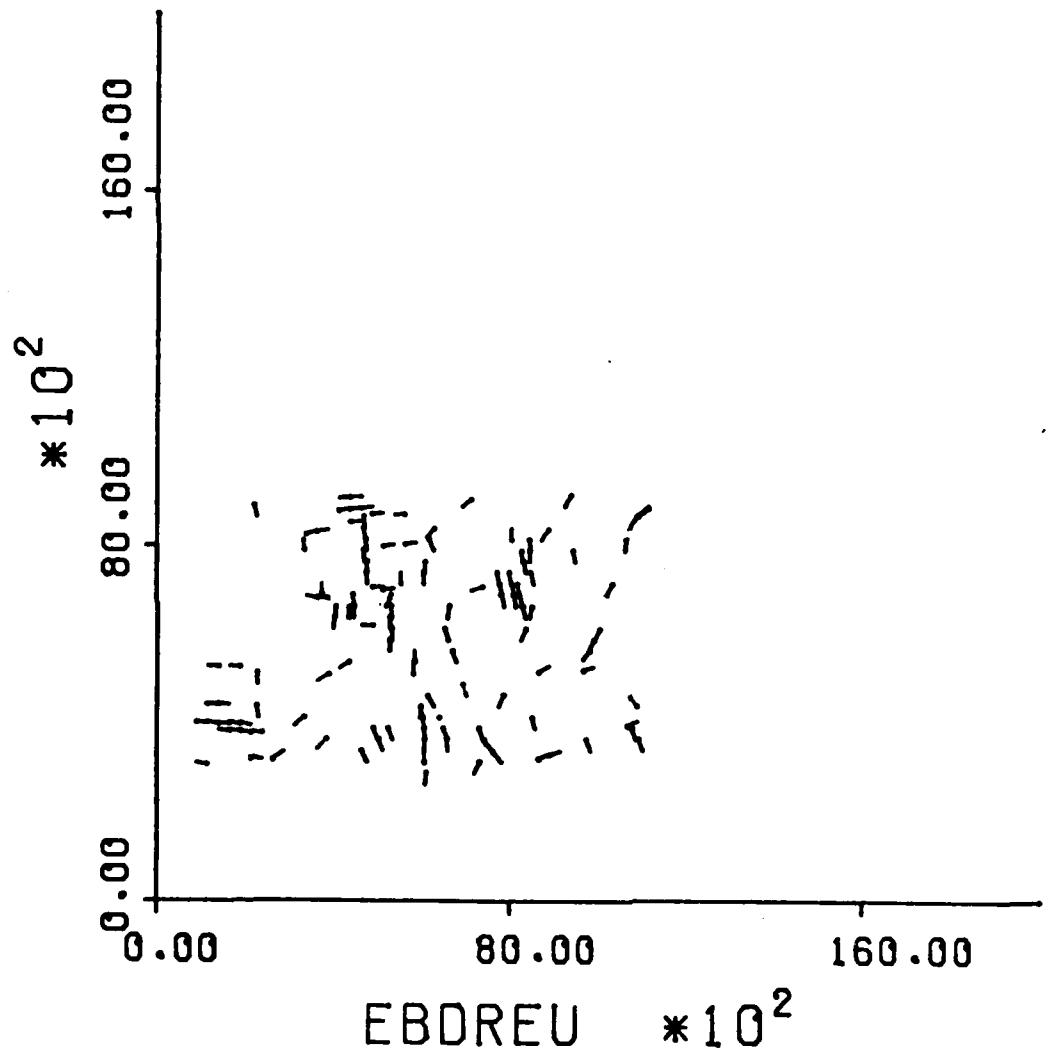


Figure 2.9(b) Plot of straight image edges extracted from DREUX 13.

Table 2.1

Details of primitive edge extraction experiments  
using 4621 and Dreux 13 (Program EDGEX.)

window size	32x32 pixels
window spacing	250 stagels
# points selected	5%
min gradient threshold	8
# points on line (coarse)	20
# points on line (fine)	14
# points on curve	15

Several isolated detections that appear to be noise were made on drainage ditches or mowing lines in fields. 199 detections were made overall; roughly half by the Hough detector.

While the primitive edges from 4621 produce a poor rendition of 4621 image structure, they nevertheless permit human recognition of the region portrayed. This is not true of edge elements extracted from the Dreux image and plotted in Figure 2.9(b). The boundaries between a few fields are nicely covered but most of the detections are isolated from others so that the human eye does not see global linearity; for instance the roads are not apparent due to their change of direction and spotty coverage. 262 detections were made overall, about half by the Hough detector and half by the curve linking procedure. However, despite the lack of structure evident to a human, there was ample structure for the automatic registration procedure to align the image with a map of the area. These registration results are presented in Section 3. Also, the primitive edge elements were used as input by another procedure which extended the edge segments and searched for intersections along them as described later.

## 2.4 Using ROSA Frequency Domain as Edge Detector

An experiment was performed to see if the ROSA frequency domain detector could find edges as well as the Hough edge detector. The Hough detector had previously been applied to the image 4621 and the positions of the extracted edges were recorded. Fifty of these detections were randomly selected for sampling by the ROSA detector. Thirty control positions, where the Hough detector found no edges, were also selected for sampling. The 32 wedge signatures for each sample were examined in order to find a scheme for determining whether an edge existed in the sample.

The first, and as it turns out, the best classification scheme was to use the ratio of the maximum wedge value to the minimum wedge value. If this ratio was less than 2.5, then the sample was said to contain no edges; if the ratio was greater than 2.5, then the sample was said to contain an edge. Using the Hough detector as a standard, this technique erroneously classified one non-edge sample as an edge sample. The histogram of the ratios of the two groups is shown in Figure 2.10(a).

More elaborate classification techniques were also tried, but none of these worked as well. For example, Figure 2.10 also shows the histograms of the ratios of the maximum wedge value divided by the median wedge value, the upper octile wedge value divided by the median wedge value, and the upper quartile wedge value divided by the median wedge value. None of these methods separated the two groups as well as the first method did.

Other ratios such as dividing the upper quartile, upper octile, and the median by the maximum were tried. In addition, comparing the percentage difference between these three ratios were also examined. These methods separated the groups even more poorly.

Since computing the ratio between the maximum and minimum wedge value is

very simple, and the position of the maximum wedge determines the orientation of the edge, it is felt that the ROSA detector could efficiently be used as an edge detector.

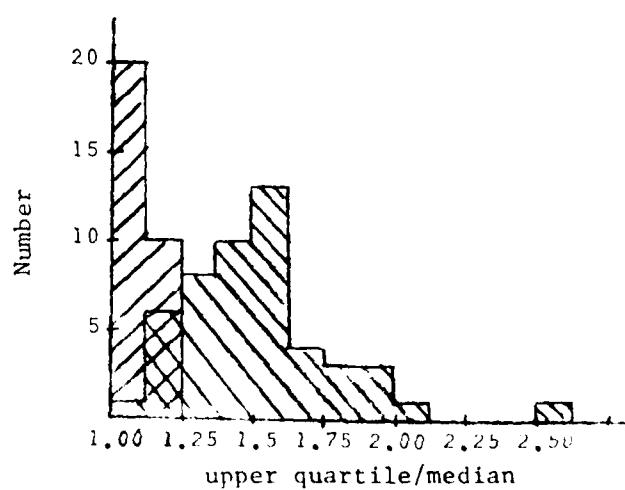
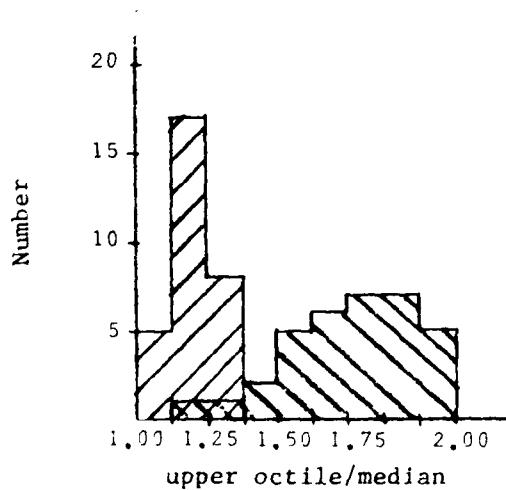
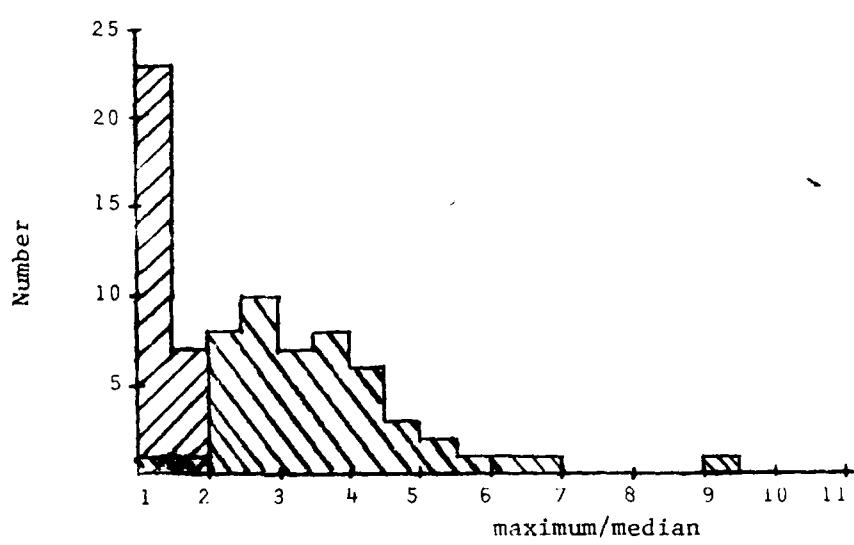
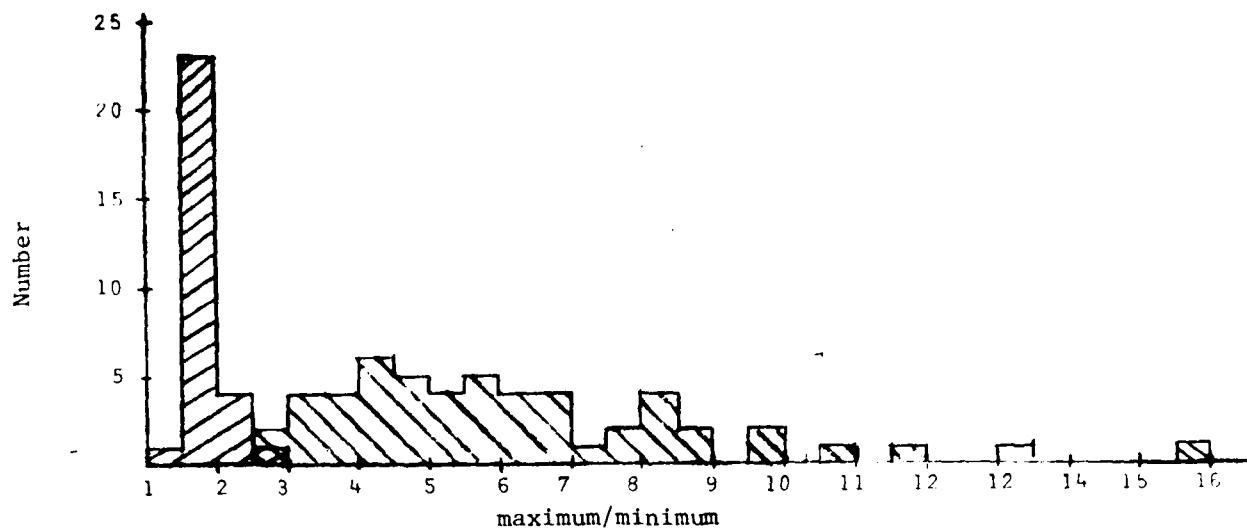


Figure 2.10. Histograms of various classification schemes to differentiate between a non-edge group (lower shown) and edge group (upper group shown).

Edge extension

Edges sometimes fade away and sometimes continue to meet other edges. For instance, a road may end in a field but is perhaps more likely to end by meeting another road. Once a straight edge segment is detected, it is therefore wise to attempt to track its full extension. A program was written to attempt to iteratively extend a detected straight edge in both directions. (The program is EDGEY and is detailed in Appendix C.) In attempting the extension, all competing gradient directions are masked off so that the edge is not lost in a background of competing edges. The image window is positioned so that the predicted extension segment would pass through its center: thus any detection should have  $R = 0$ .

Primitive detections in the 32x32 window represent only 1/8 inch on the actual film. In some cases documented in section 2.7 extensions of up to an inch or more were made. It is possible to extend this technique so that it could be applied to curved edges. While straight edges are extended or are found to terminate, it is appropriate to check for intersecting edges as described in the next section.

Edge intersection points are highly desirable as photo pass points and are thus a target for automatic recognition effort. Moreover, we would like to describe the intersection as much as possible so that it can be differentiated from other intersections. Figure 2.11 shows a catalog of possible intersection types which we might like to recognize automatically. It was decided that first an algorithm should be developed to detect arbitrary intersections, then further testing could be used to classify the detected intersection. Toward this end a simple routine was implemented to test for edge activity nearly perpendicular to an existing edge. In fact, intersections would be checked for while extending existing edges. (Program 'EDGEY again). Each existing edge element is placed into one of 4 classes as shown in Figure 2.12. Then the window can be positioned once on each side of the edge so that the Hough detector can detect intersecting edges. Of course the current edge direction as well as its  $180^{\circ}$  supplement are masked off during detection. The two window placements are made every 100 or so stages along the existing edge so that no neighboring region is ignored.

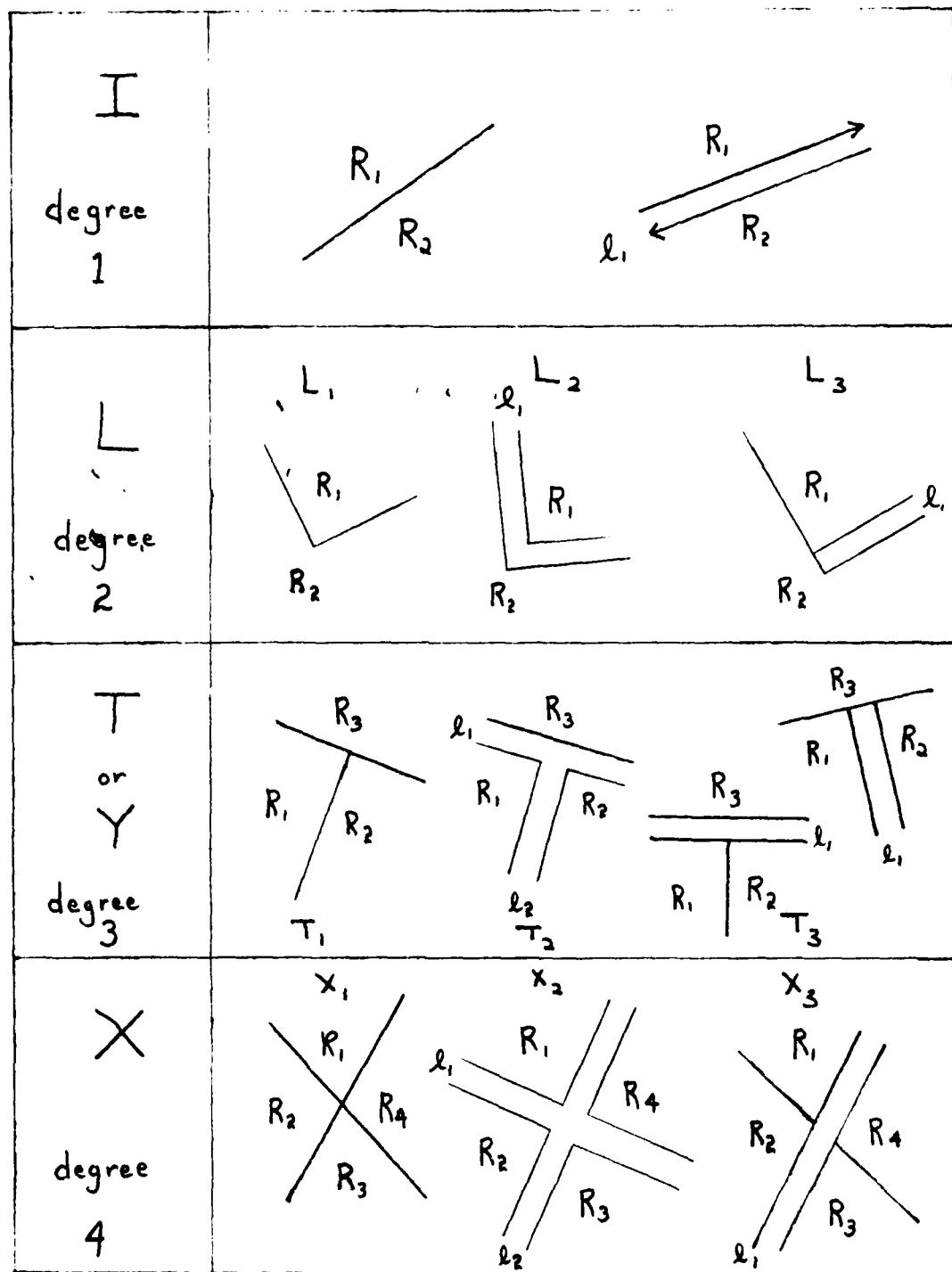


Figure 2.11 Catalogue of possible local elementary features for recognition & registration.

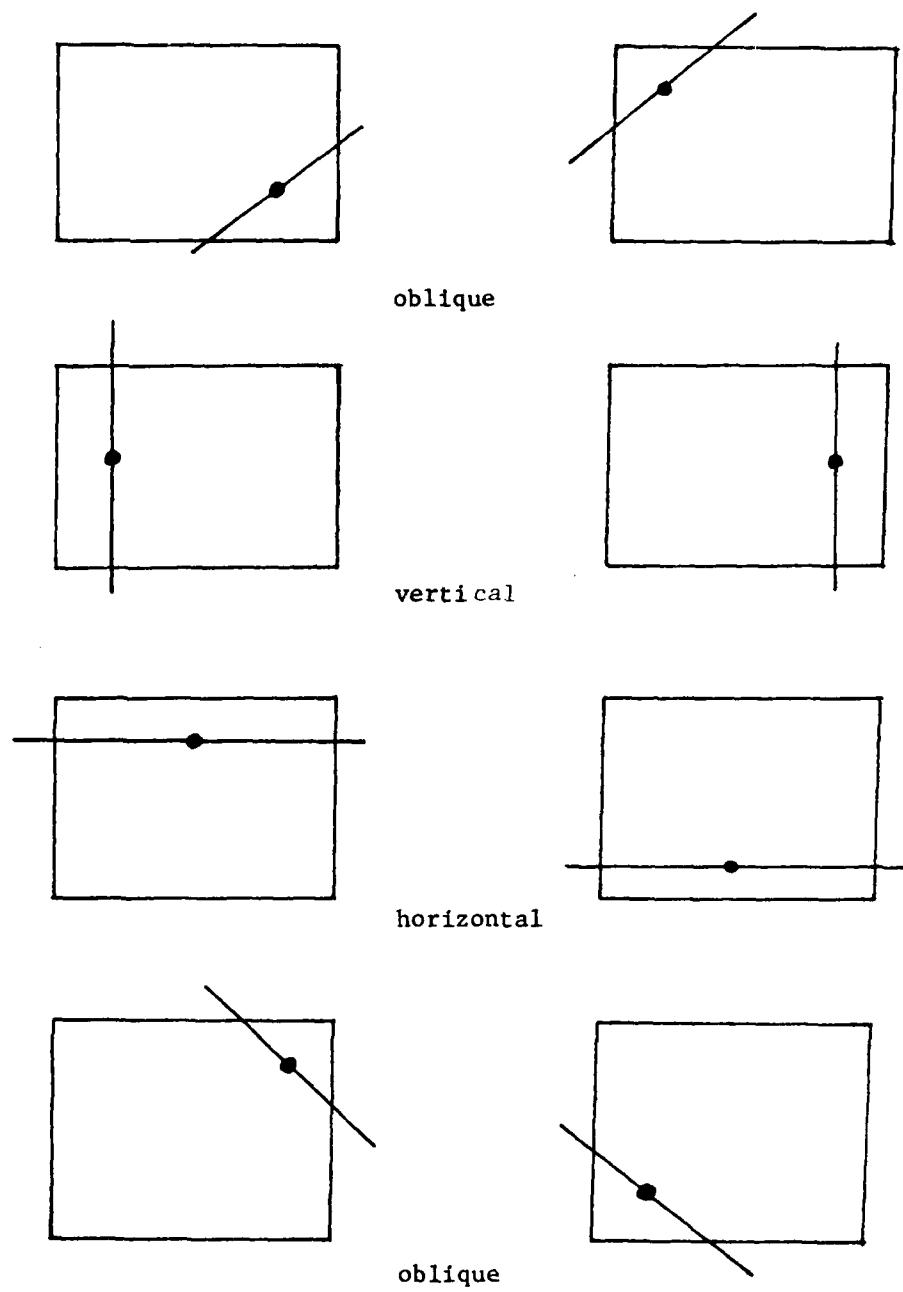


Figure 2.12

Window positions used to search for intersecting edges. Existing edge elements are placed in one of 4 classes and the search window positioned accordingly.

### 2.6.1 Experiments with edge extension and intersection detection.

In the first experiment with intersection detection, the gradient masking technique was tested to see if masking of a major edge would allow detection of weaker intersecting edges. Certain lineal features of 4621 are shown in Figure 2.13. Several search tracks along certain of these lineals were selected for intersection detections.

Searches were conducted between the following pairs of points:  
-- point 1 to point 6, point 6 to point 11, point 8 to point 21,  
point 1 to point 29, point 3 to point 14, and point 15 to point 16.  
Windows were sampled at intervals of 150 stagels along each lineal segment and on both sides of the lineal as shown in Figure 2.12.  
For all windows the two gradient directions normal to the track were masked off to remove contention from the major edge points. If a detection was made on one side only, then a "1" intersection was reported.  
If a detection was made on both sides, an "X" intersection was reported.

Results of intersection detection are given in Table 2.2 .  
Some of the desired connecting roads and driveways were detected (i.e. at points 1,3,4,5,6,8,11 and 17) and some were missed (i.e. at points 2 and 9). Many field boundaries were picked up as well as some driveways not numbered in Figure 2.13.

The program was modified to extend edges and was run on the DREUX 13 data. The results are plotted in Figure 2.14. Detected intersections are indicated by the small circles. Extended edges are also evident in the plot. Some edge elements were lengthened from 1/8 inch to an inch or so.

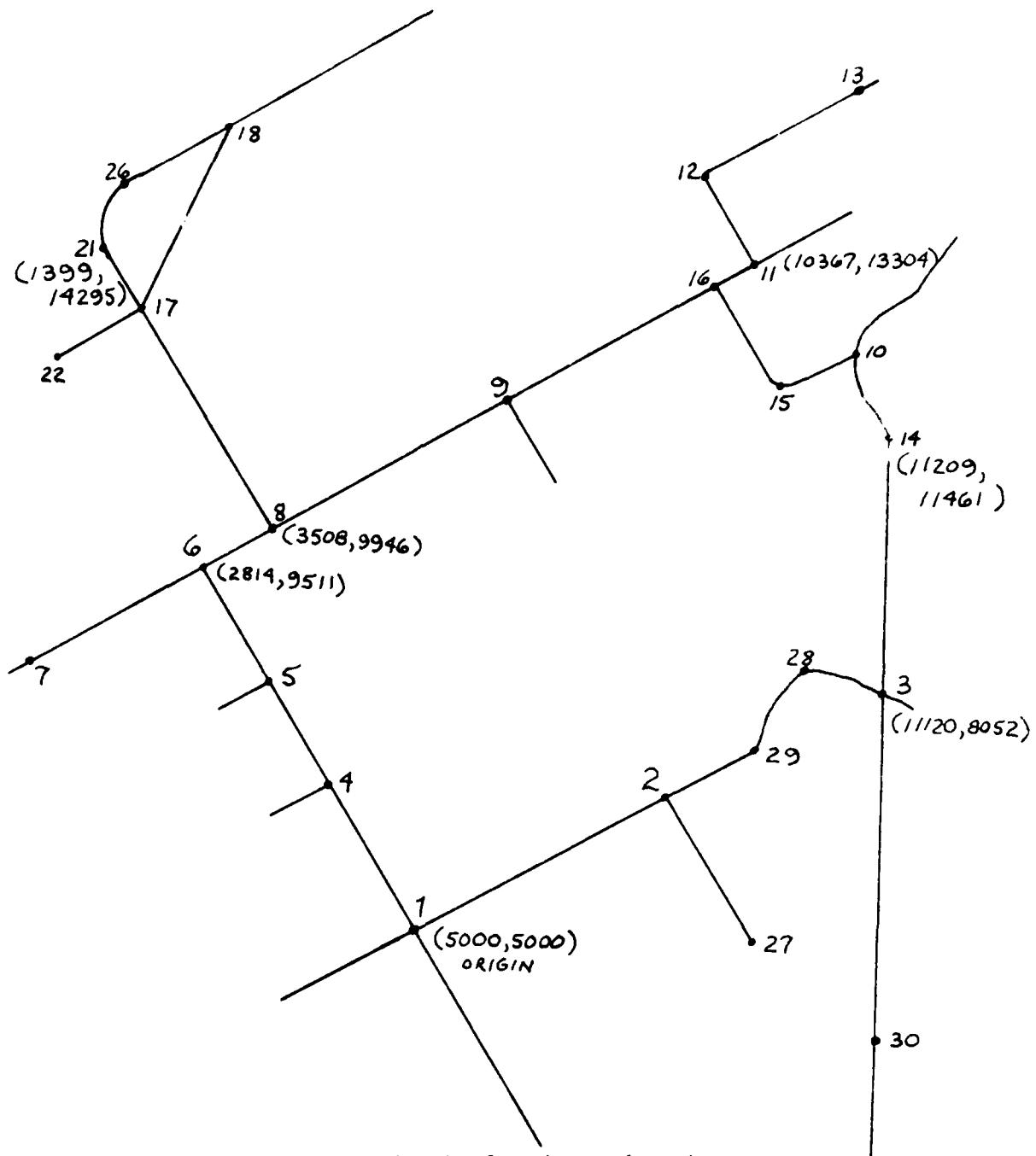


Figure 2.13

Rough sketch of paths used on image 4621 to search for intersecting edges. (Coordinates are stage coordinates relative to origin at (5000,5000)).

Table 2.2  
Results of Connecting Edge Experiment on Image 4621

Intersection in Stage Coord.	Type of Intersect.	Description
tracking from point #1 to point # 6 steps of 150 stagels		
(4999,5011)	T	point #1 (really an X)
(4937,5137)	T	road/field
(4850,5317)	T	road/field
(4828,5362)	T	road/field
(4691,5642)	T	road/field
(4499,6032)	X	road/drainage
(3997,7060)	T	road/field
(3913,7242)	T	point # 4 road/driveway
(8762,9010)	T	field boundary/road
(3612,7849)	T	field/road
(3325,8436)	T	point # 5
(3029,9041)	T	road/field
(2893,9320)	T	? investigate
(2788,9533)	T	point # 6
tracking from point # 6 to point # 11		
(3100,9654)	T	road/field
(2888,9548)	T	point # 6
(3569,9888)	T	point # 8
(4046,10127)	T	woods?
(4698,10453)	T	road/field
(6020,11113)	T	road/driveway
(6942,11575)	T	driveway to buildings
(7063,11635)	T	buildings
(6986,11596)	T	buildings
(8276,12241)	T	road/driveway (unnumbered point)
(8705,12456)	T	road/driveway (unnumbered point)
(10480,13344)	T	point # 11
tracking from point # 8 to point # 21		
(3526,9900)	T	point # 8 again
(3240,10502)	T	road/field
(3103,10786)	T	road/field
(1756,13584)	X	point # 17
(1557,13997)	T	road/field
track from point # 1 to point # 29		
(5008,5003)	X	point # 1
(6765,5849)	T	noise, threshold set too low
(6706,5821)	T	noise, threshold set too low
(9100,6980)	T	road/field

Table 2.2 (continued)

Intersection in Stage Coord.	Type of Intersect.	Description
track from point # 3 to point # 14		
(11120,8072)	T	point # 3 (really an X)
(11147,9416)	T	road/treeline
(11153,9716)	T	road/treeline
(11158,9975)	T	road/field
(11170,10546)	T	road/treeline
(11179,10989)	T	road/field
(11190,11531)	T	?
(11191,11589)	T	actually same road bending
track from point # 15 to point # 16		
(10454,12220)	T	small driveway
(10415,12296)	T	road/trees
(10112,12892)	T	road/trees

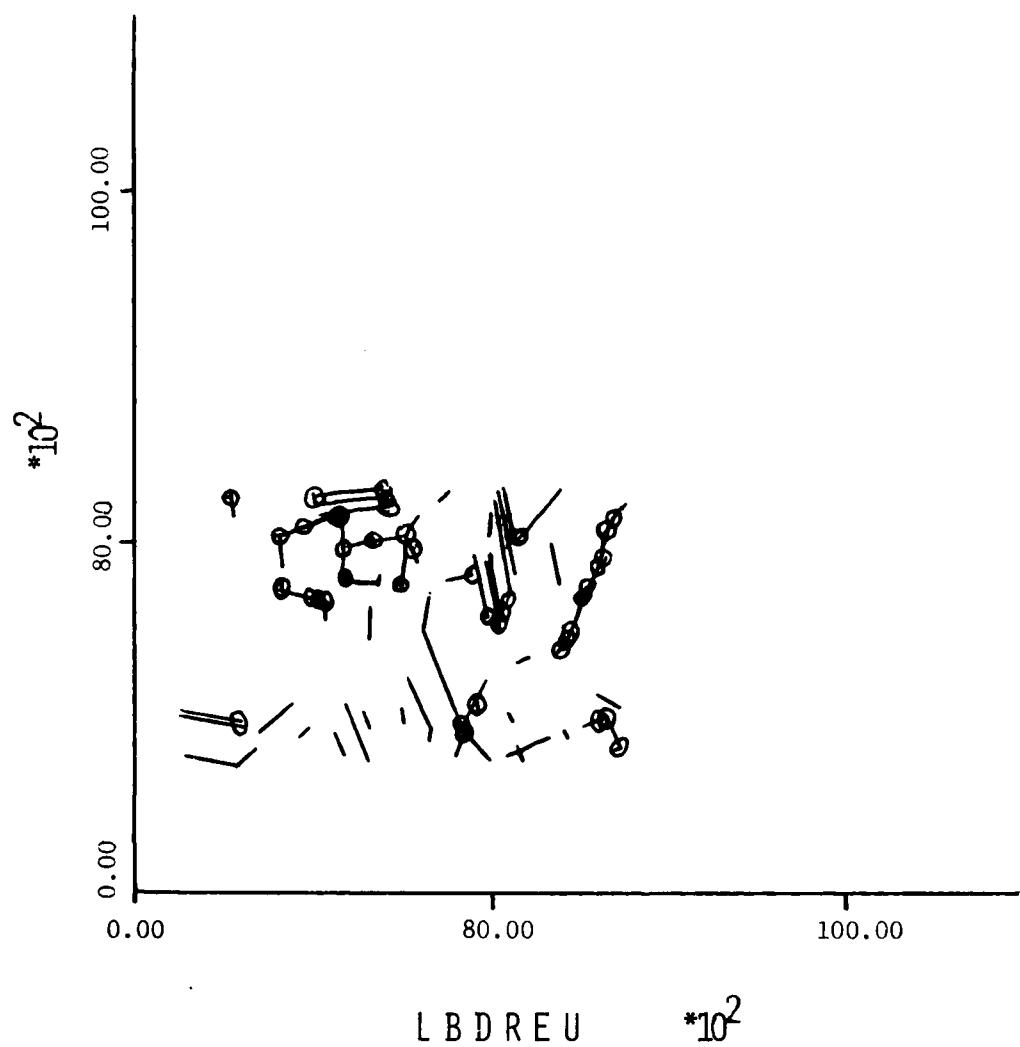


Figure 2.14 Plot of results of edge extension and intersection detection on Dreux 13. Circles indicate automatically detected intersections.

## 2.7 Intersection Classification

Apart from the work described in Section 2.6, an intensive investigation into the detection and classification of intersections given an initial set of detected line segments was conducted. This work is easily integrated into the feature detectors EDGE<sub>X</sub> and EDGE<sub>Y</sub>.

The detection and classification of intersections given a set of detected line segments was divided into three steps. The first step was to merge line segments with approximately the same orientation and position. The second step was to pair line segments which were 180° off in orientation, but very close to each other. This step would associate the two sides of a road as a single entity. The last step was to take the resulting paired line segments and leftover unpaired line segments and determine whether any of them intersected and classify the intersections as shown in Figure 2.11. The software for the classification is presented in Appendix F.

### 2.7.1 Merging Line Segments

The goal of the first step is to take a set of detected line segments and to merge line segments which have the same approximate position and orientation. The three possible types of situations where merging should be done are shown in Figure 2.15. The first case is called linking, i.e. one line segment can be extended and merged with the second. The second and third cases are called domination, i.e. where one image edge was detected twice in slightly different locations. This can happen when overlapping edge detection windows are used.

The merging algorithm steps are:

- (1) for each line segment  $v_i$  compare with each of the other line seg-

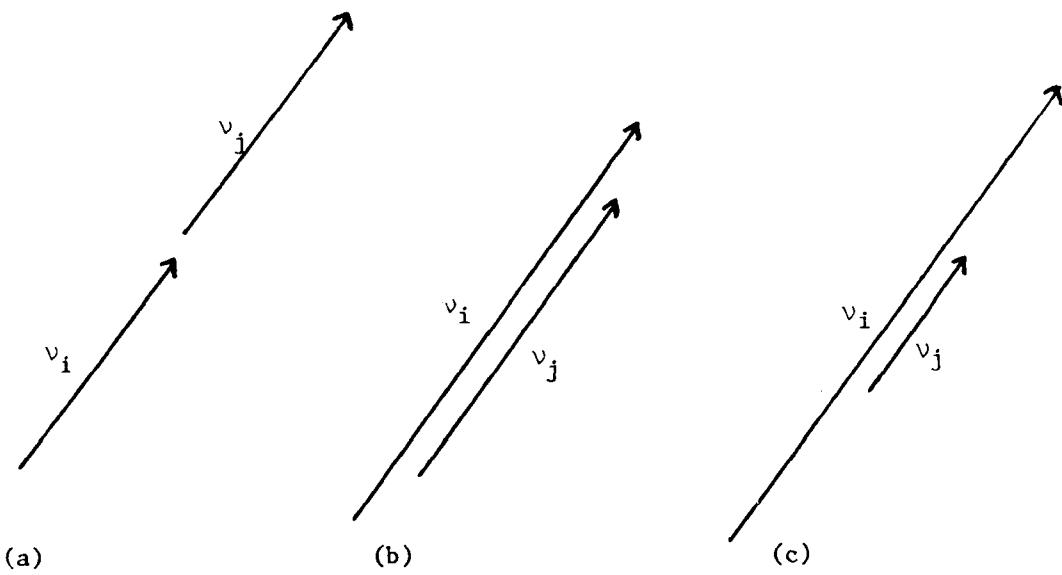


Figure 2.15. The three cases where merging of line segments should be done.

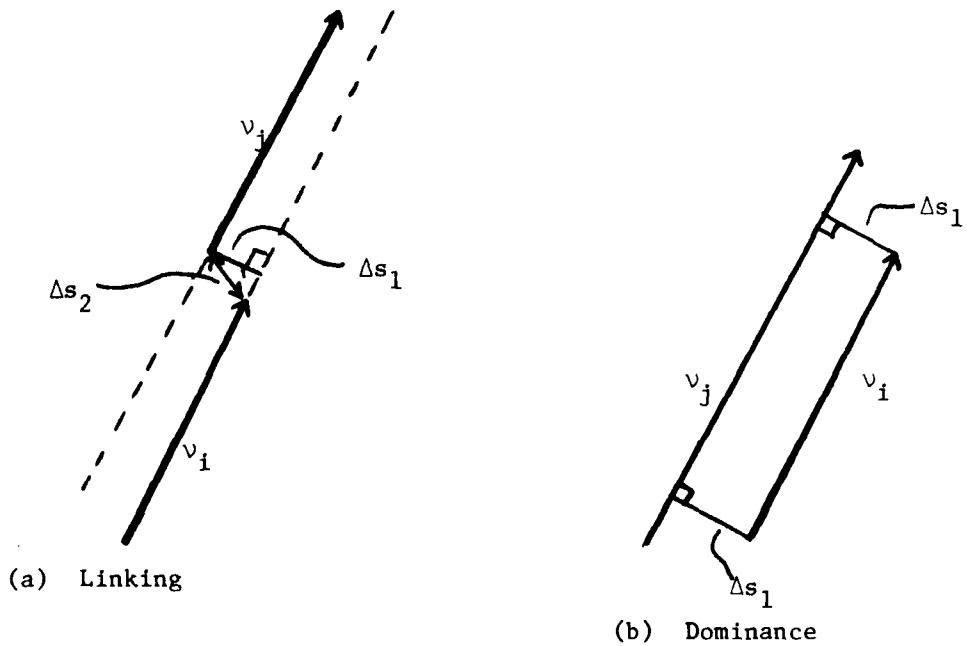


Figure 2.16. In order for merging to occur, the indicated distances must be less than a supplied tolerance.

ment  $v_j$  in order of increasing distance between  $v_i$  and  $v_j$ . This ordering is determined by calculating the  $(R, \theta)$  form of the line segment [see Duda and Hart, (1972) Hough edge detector] and computing the differences between the polar radii,  $R$ , of each pair. The comparison is done with closest line segments first, since the result of combining three or more edge segments was dependent upon the order of combination.

- (2) For each pair  $(v_i, v_j)$ , it was determined if linking or domination would be needed. For linking to be indicated (a) the angle between  $v_i$  and  $v_j$  had to be less than  $\Delta\theta$  tolerance, (b) the distance between the endpoints and the other line had to be less than  $\Delta s_1$ , and (c) the linking separation had to be less than  $\Delta s_2$ , as shown in Figure 2.16 (a). If domination was indicated, then, in order for the pair to be merged, the distance between the endpoints and the other line had to be less than  $\Delta s_1$ , as shown in Figure 2.16 (b).
- (3) Finally if the line segment pair satisfied the merging test, they were merged and replaced with the new merged line segment.

Several methods of determining the endpoints of the new merged line segment, for the cases shown in Figure 2.15 (a) and (b), were tried. The final method found the slope of the new line by averaging the slope of the two old segments. A point on the new line was found by averaging the endpoints of the two old line segments. The endpoints of the new line segment were calculated using the extremes of the old line segments. Note that, in averaging two almost vertical lines whose slopes are opposite in sign, the average slope should tend towards infinity instead of zero. This was accomplished by averaging the inverse of the slopes and then taking the inverse of the re-

sult. The extremes used to find the new endpoints were either the extreme x-coordinate values, if the slope was less than 1.0, or the extreme y-coordinate values, if the slope was greater than 1.0. This endpoint procedure was used because small differences in x-values can lead to large differences in y-values for nearly vertical lines and similarly for nearly horizontal lines.

For the case of complete domination, shown in Figure 2.15 (c), the line segment  $v_j$  would be completely discarded in favor of  $v_i$ . That is, the new line segment would be the old  $v_i$ .

### 2.7.2 Pairing Line Segments

The goal of the second step of intersection classification and detection is to pair all possible line segments. This step would associate the two sides of a road, for example, as a single entity. The pairing was accomplished by comparing each pair of line segments  $(v_i, v_j)$  such that (see Figure 2.17):

- (1) The difference between the polar angles of the two line segments had to be within  $180^\circ \pm \Delta\theta$  tolerance,
- (2) the distance between the endpoints and the other line had to be less than  $\Delta s_3$ ,
- (3) the line segments had to overlap at least by  $\Delta o$ , and
- (4) the line segments did not intersect.

After this process, it is possible to have the situation shown in Figure 2.18, where  $v_j$  and  $v_j'$  were not merged because they were too apart, but because of their pairing with  $v_i$ , they should be merged. This last step was introduced because the intersection of two roads would be represented by the intersection of two pairs of line segments - making detection easier.

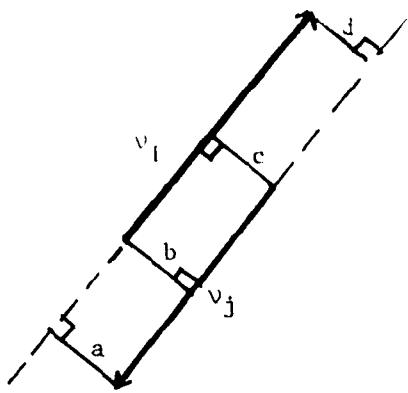


Figure 2.17. In order for  $v_i$  and  $v_j$  to be paired, the distance  $a$ ,  $b$ ,  $c$ , and  $d$  must be less than the supplied tolerance  $\Delta s_3$ .

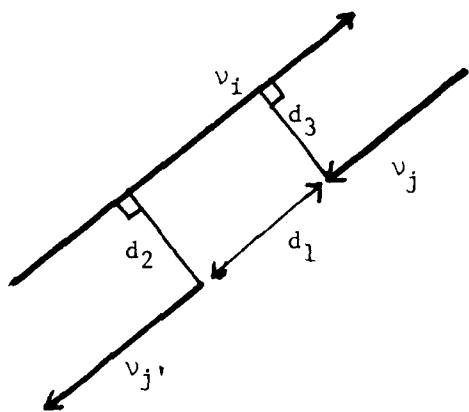


Figure 2.18. A situation where  $v_j$  and  $v_j'$  should be merged and the result paired with  $v_i$ .

Referring to Figure 2.18, the conditions on merging  $v_j$  and  $v_j'$  were:

- (1) the difference in distances  $d_2$  and  $d_3$  had to be within  $\Delta s_1$ ,
- (2) the distance  $d_1$  must be less than  $2 * \max(d_2, d_3) * \Delta s_1$ , and
- (3) the line segments  $v_j$  and  $v_j'$  have to be on the same side of  $v_i$ .

Pairs were typed as dark lines on a light background or light lines on a dark background.

### 2.7.3 Detecting and Classifying Intersections

The last step is to detect and classify intersections according to Figure 2.11. This step is accomplished by treating each pair of line segments or an unpaired line segment as one unit and comparing all such units.

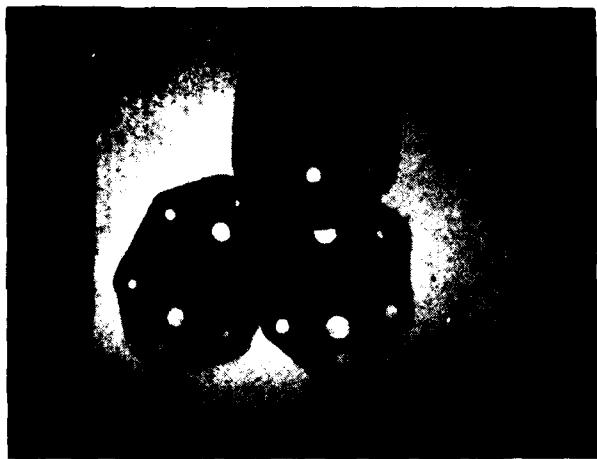
For each combination of units:

- (1) determine if they intersect or would intersect if their lengths were extended by  $\Delta s_1$  at each end, and the angle between the intersecting units is not within  $0^\circ \pm \Delta\theta$ .
- (2) if they pass (1), then the intersection is typed as 1, 2, or 3 depending on where the intersection occurred. If the intersection occurred within  $\Delta s_1$  of the ends of both units, then the type is 1; if it occurred within  $\Delta s_1$  of the end of one unit, the type is 2; else it is type 3. Types 1, 2, and 3 correspond to L, T, and X intersections, respectively.
- (3) Finally if they satisfied (1) and (2), the intersections are finally classified as  $L_1$ ,  $L_2$ , etc. depending on how many line segments are in the units. For example if two pairs are involved in a type 1 intersection then the intersection is classified as  $L_2$ .

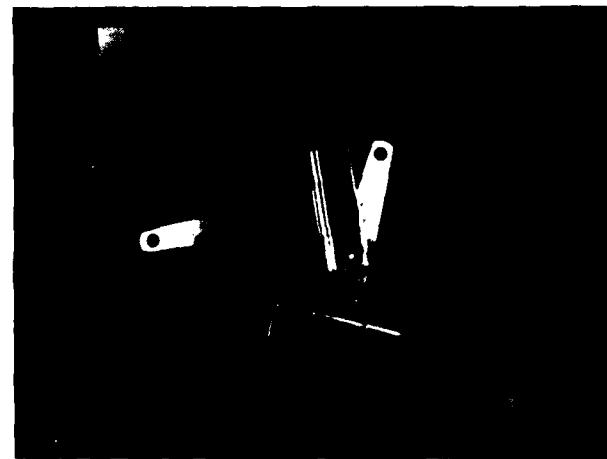
#### 2.7.4 Experiments on Detection and Classification of Intersections

The three steps of detecting and classifying intersections were applied to a series of sets of extracted line segments from images of hinges and carburetor covers. These images were selected both because the edges were already available, and they contained many straight edges. The images used are shown in Figure 2.19. The sets of extracted edges are shown before and after merging in Figures 2.20-2.24. The lists of paired line segments and their types, and the detected intersections with their types are also shown. The detected intersections are circled on the images. The values used for the tolerances  $\Delta s_1$ ,  $\Delta s_2$ ,  $\Delta s_3$ ,  $\Delta\theta$ , and  $\Delta o$  were 7 pixels, 7 pixels, 25 pixels, .75 radians, and 50% respectively. Figure 2.25 shows a test data set demonstrating the capability of the algorithm.

The results shown indicate that intersections can be detected and classified automatically. This algorithm can easily be combined with EDGEY to include exploration for missing line segments in the image. Automatic detection and classification of intersections are important steps in the LNK registration procedure described in the next section.



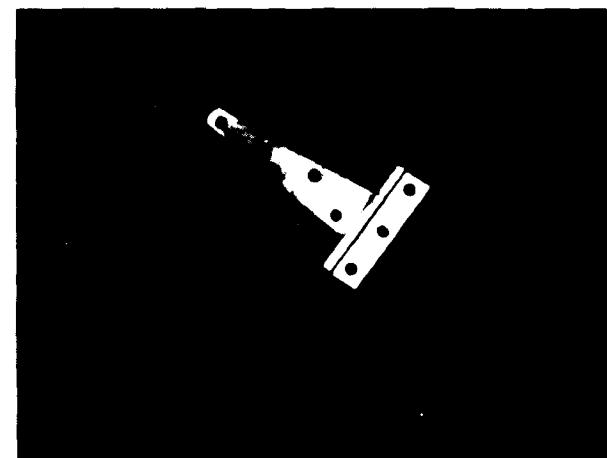
(a)



(b)



(c)



(d)

Figure 2.19 Images used to test detection and classification of intersection algorithm.

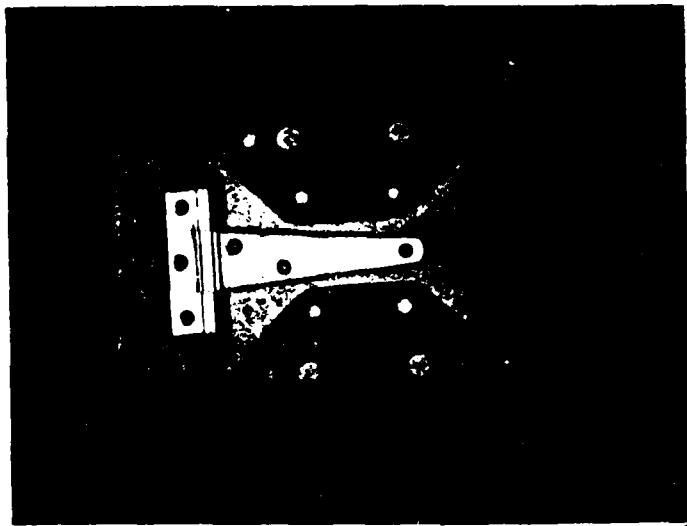
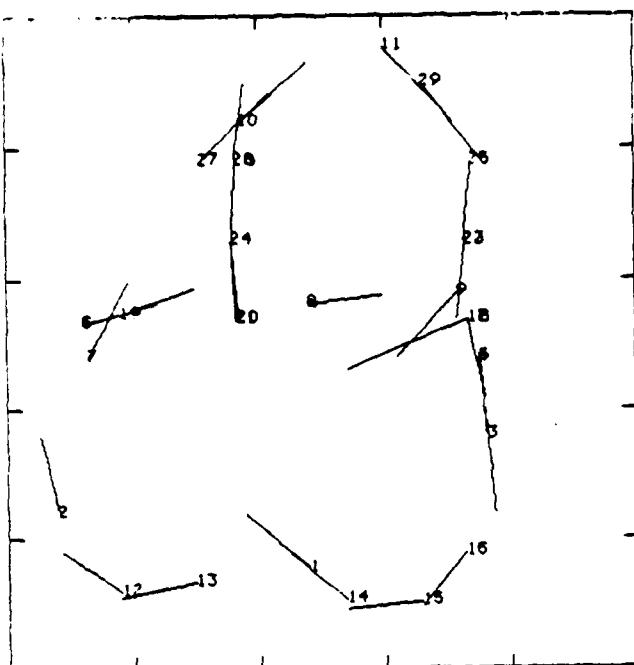


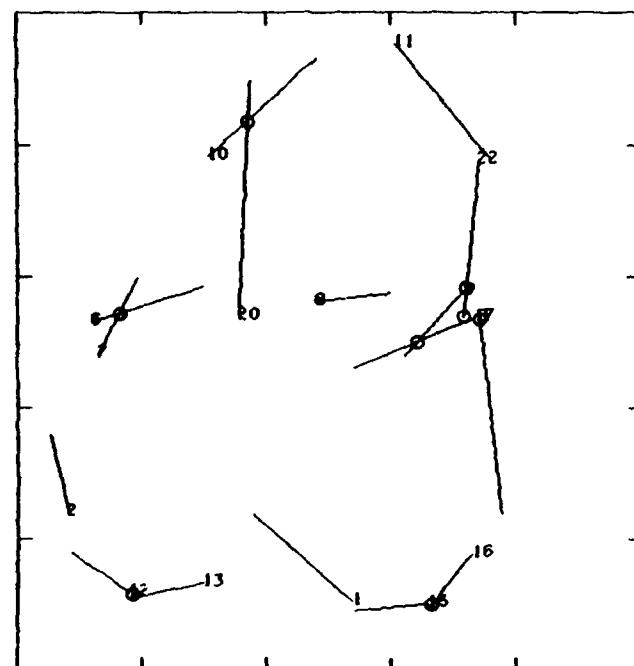
Figure 2.19 continued



(a)

PAIR LIST

None

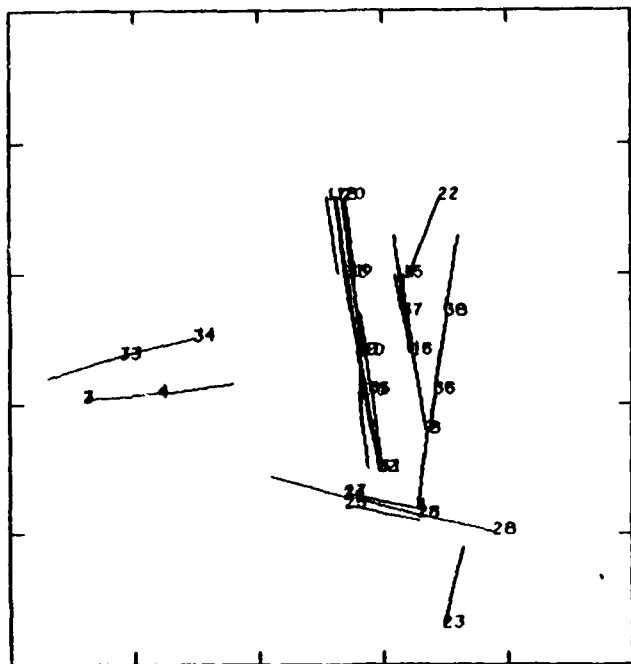


(b)

INTERSECTION LIST

<u>Edges</u>	<u>Type</u>
3 17	L1
6 7	X1
9 17	X1
9 22	T1
10 20	X1
12 13	L1
15 16	L1
17 22	T1

Figure 2.20 (a) Original edge segments extracted from image in Figure 2.19(a). (b) Result of applying merging algorithm. The detected intersections are circled.

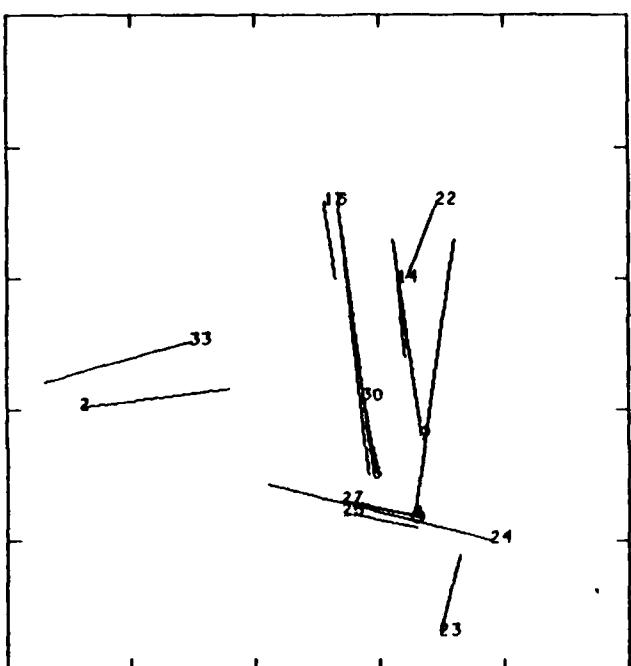


(a)

PAIR LIST

<u>Edges</u>	<u>Type</u>
5	17
24	25
24	27

Dark on Light  
Dark on Light  
Light on Dark



(b)

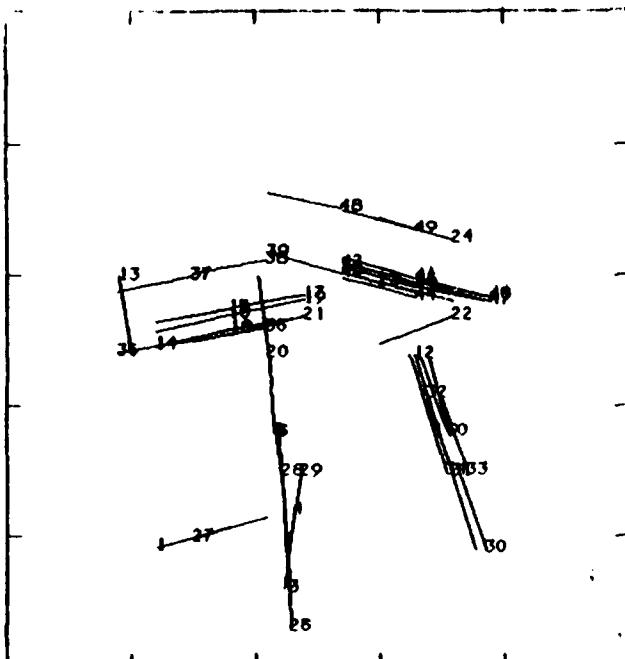
INTERSECTION LIST

<u>Edges</u>	<u>Type</u>
1	24 25
1	24 27

T3  
T3

Figure 2.21

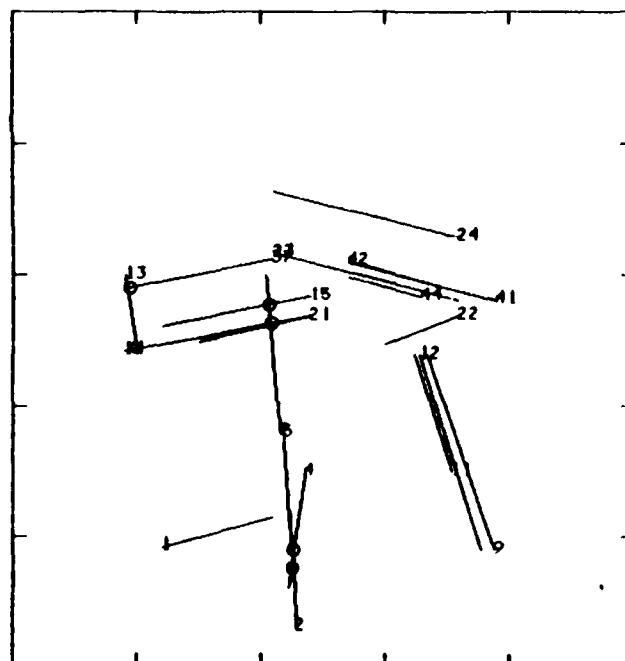
(a) Original edge segments extracted from image in Figure 2.19(b). (b) Result of applying merging algorithm. Detected Intersections are circled.



(a)

## PAIR LIST

<u>Edges</u>		<u>Type</u>
9	12	Dark on Lig <sup>!</sup>
11	12	Light on Dar <sup>!</sup>
14	15	Dark on Lig <sup>!</sup>
14	21	Light on Dark
23	41	Dark on Lig <sup>!</sup>
23	44	Light on Dark
42	44	Light on Dark

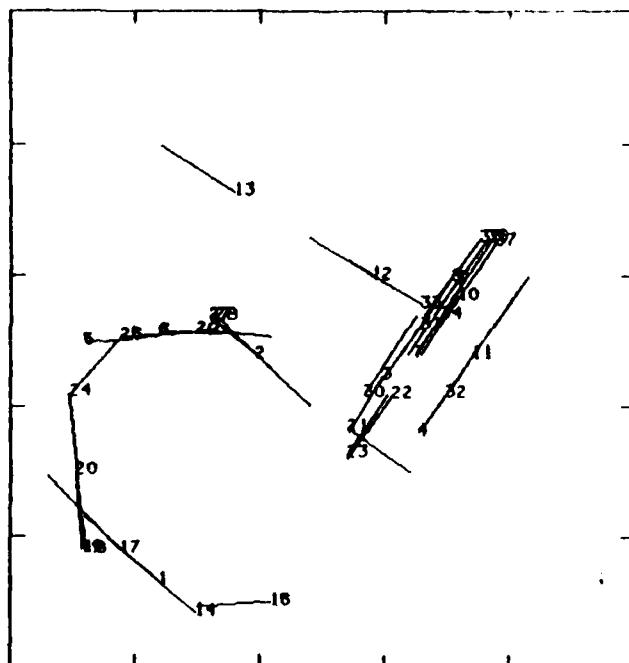


(b)

## INTERSECTION LIST

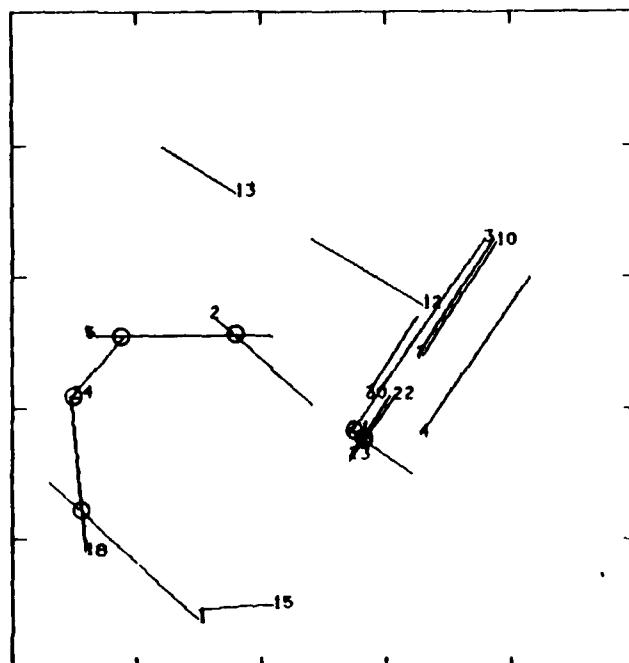
<u>Edges</u>	<u>Type</u>
2 4	X1
4 5	T1
7 14 15	X3
7 14 21	X3
13 14 15	T3
13 14 21	T3
13 37	T1

Figure 2.22 (a) Original edge segments extracted from image in Figure 2.19(c). (b) Result of applying merging algorithm. Detected Intersections are circled



(a)

## PAIR LIST

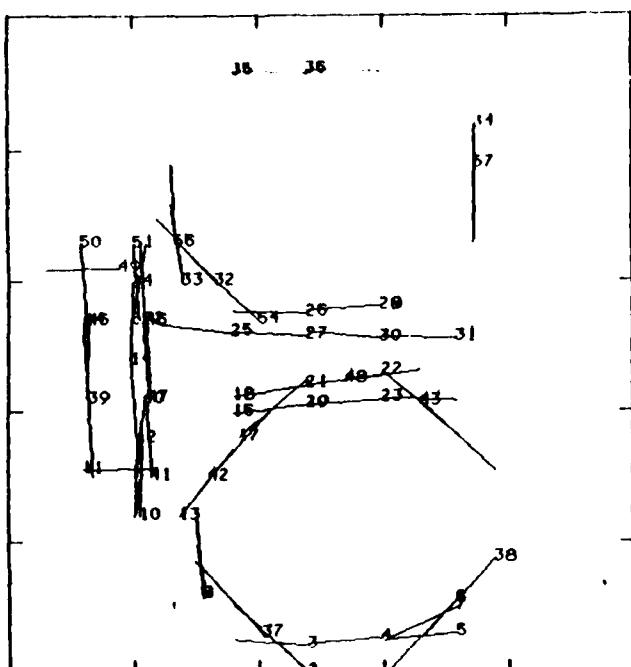


(b)

### INTERSECTION LIST

<u>Edges</u>		<u>Type</u>
1	18	X1
2	5	X1
3 7	21	L3
3 23	21	L3
3 30	21	L3
5	24	T1
18	24	L1
21	22 23	X3

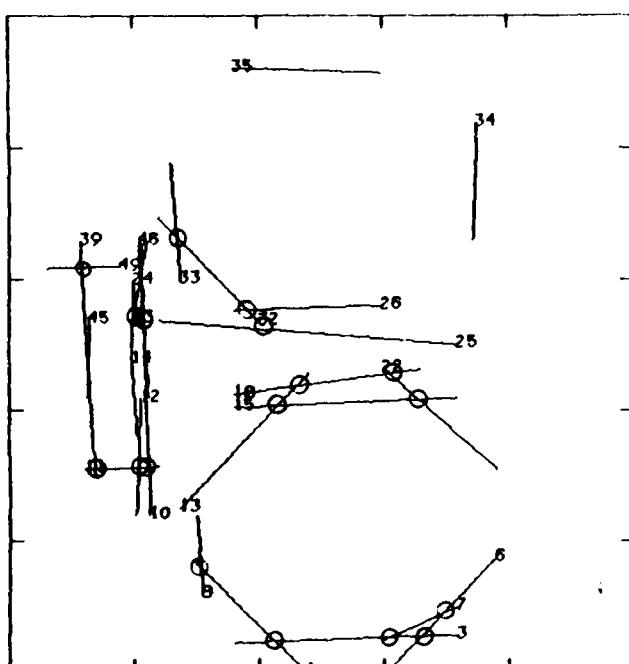
Figure 2.23 (a) Original edge segments extracted from image in Figure 2.19(d). (b) Result of applying merging algorithm. Detected intersections are circled.



(a)

### PAIR LIST

Edges	Type
10 12	Dark on Light
10 14	Dark on Light

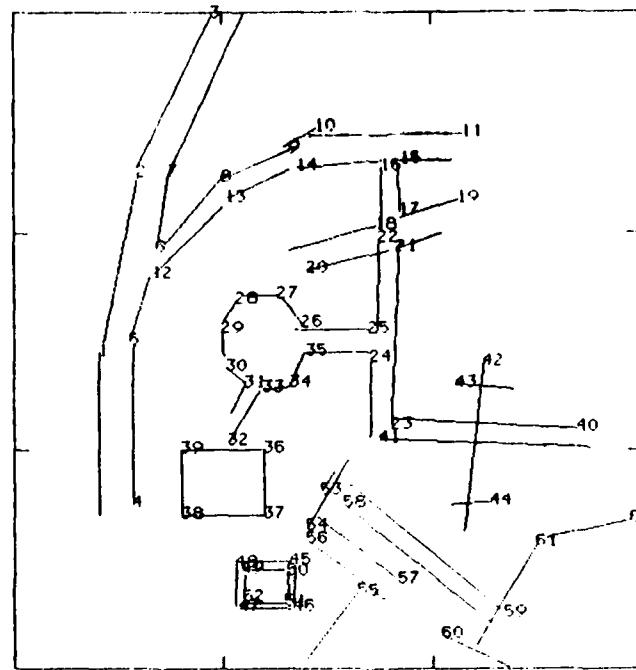


(b)

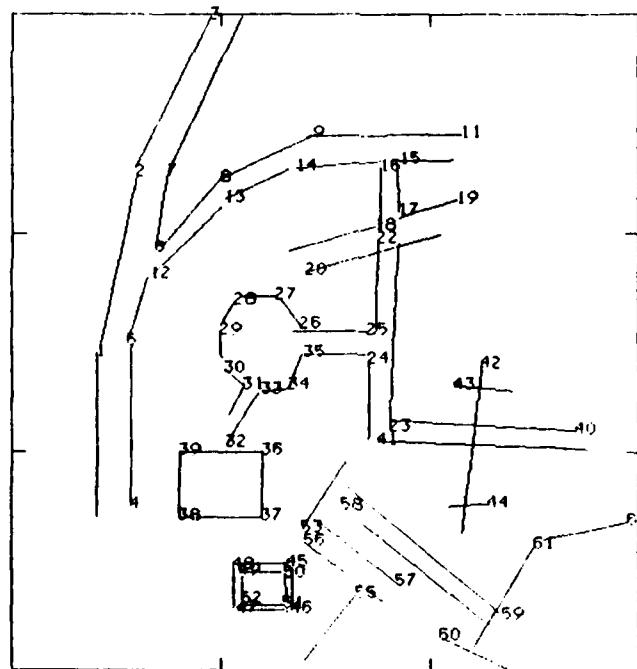
#### INTERSECTION LIST

<u>Edges</u>		Type	
1	3	X1	
1	8	T1	
3	6	X1	
3	7	T1	
6	7	X1	
10	12	11	X3
10	12	53	X3
10	14	11	X3
10	14	53	X3
11	39	T1	
13	15	X1	
13	18	X1	
15	22	X1	
18	22	T1	
25	32	T1	
26	32	X1	
32	33	X1	
39	49	X1	
46	53	X1	

Figure 2.24 (a) Original edge segments extracted from image in Figure 2.19(c). (b) Result of applying merging algorithm. Detected intersections are circled.

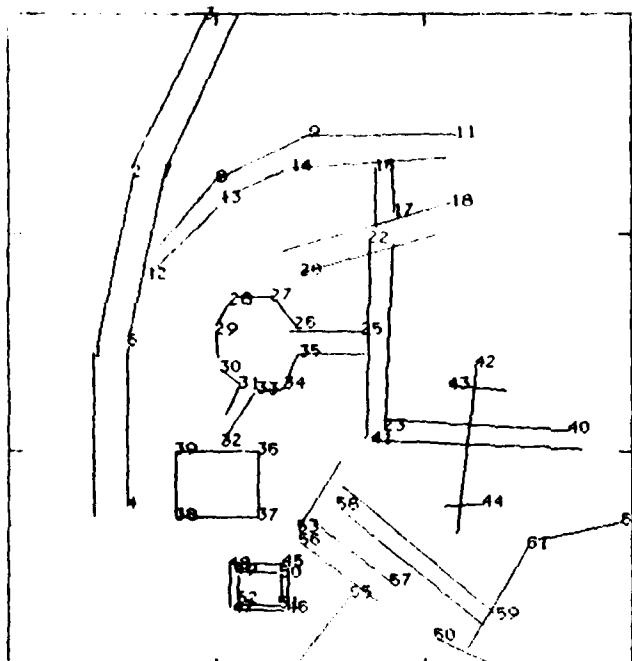


(a)



(b)

**Figure 2.25** Test image showing some of the capabilities of the algorithm. The original input set is shown in (a). The result of the merging step is shown in (b).



AIR LIST

<u>Edges</u>		<u>Type</u>
1	4	Dark on Light
2	5	Dark on Light
58	59	Dark on Light
3	7	Dark on Light
4	39	Light on Dark
8	12	Dark on Light
9	13	Dark on Light
11	14	Dark on Light
57	58	Light on Dark
16	17	Dark on Light
18	20	Dark on Light
56	57	Dark on Light
22	23	Dark on Light
50	52	Light on Dark
25	35	Dark on Light
31	32	Dark on Light
38	45	Light on Dark
40	41	Dark on Light
40	43	Light on Dark
45	47	Dark on Light
45	49	Dark on Light
46	50	Dark on Light
47	51	Dark on Light
48	52	Dark on Light
49	51	Light on Dark

#### INTERSECTION LIST

Edges				Type
1	4	2	5	L2
2	5	3	7	L2
2	5	8	12	T2
8	12	9	13	L2
9	13	11	14	L2
11	14	16	17	T2
16	17	18	20	T2
18	20	22	23	T2
22	23	25	35	T2
22	23	40	41	L2
22	23	40	43	L3
25	35	26		L3
25	35	34		L3
26		27		L1
27		28		L1
28		29		L1
29		30		L1
30		31	32	L3
30		33		L1
31	32	33		L3
31	32	36		T3
33		34		L1
36		37		L1
40	41	42		X3
40	43	42		X3
42		44		X1
45	49	46	50	L2
45	49	48	52	L2
46	50	47	51	L2
47	51	48	52	L2
53		56	57	L3
53		57	58	L3
53		58	59	L3
55		56	57	T3
57	58	61		T3
58	59	61		T3
60		61		T1
61		62		L1

Figure 2.25 The result of the pairing step is shown in (c). Note that some edges were merged that were not merged in (b). The pair list and intersection list are also shown. The intersection list was edited to remove duplicate detections.

## 2.8 Region Segmentation

A segmentation of an image is a partition of the image into disjoint subsets whose union is the entire image. For surveys of segmentation see [Riseman and Arbib 1977, Kanade 1980, Zucker 1976, Pavlidis 1977]. Many segmentation procedures produce an initial segmentation and then apply an iterative procedure such as merging or splitting to obtain an improved segmentation. Two basic approaches to finding an initial segmentation are first locating boundaries or first locating pixels with similar feature values.

Boundary detection generally consists of edge detection followed by linking of edges into closed boundaries as discussed in Section 2.2. Edge detection procedures which form closed edges such as the Marr detector [Marr 1979] and relaxation labelling [Zucker 1977] provide a segmentation directly. Edge detection in textured images is a difficult task which depends heavily on the relative scale of the texture and the regions' sizes.

Construction of regions based on similarity of pixels or pixel neighborhoods requires feature extraction. Features commonly measured include average gray level in a neighborhood, variance, average edge content per unit area, average orientation of local edges and average spot size of uniform contiguous areas. Texture measures, such as co-occurrence matrices and Fourier transform ring data can be used as feature measures for larger areas. Threshold techniques [Pavlidis 1977, Price 1976] can be used with these features to provide an initial segmentation.

Milgram [1978] describes a procedure for region construction using evidence from several sources such as edge information and pixel feature values. The algorithm selects the contours at different thresholds according to the support of the edge data along the contours. Zucker [1979] gives a relaxation technique for constructing regions from primitive edges. This scheme

allows points to be considered as interior or boundary points of a region.

The relaxation process allows edge segments separating regions to prosper as region points and edge points reinforce themselves.

Pavlidis describes a general class of split-merge algorithms using the notion of region adjacency. Given a criterion for deciding if a single region should be split and a criterion for deciding whether two adjacent regions should be merged, the procedure is as follows:

- 1) Split each region which should be split according to the splitting criterion. Continue this until no further regions satisfy the splitting criterion.
- 2) If step 3 has not been executed yet then keep going, else if no merges occurred in the last execution of step 3, then stop.
- 3) Merge any two adjacent regions which should be merged according to the merging criterion. Continue this step until no adjacent regions satisfy the merging criterion.

### 3.0 Registration of image data to a map.

The concept of registration is crucial to image interpretation. Very generally it applies to matching images from different sensors, matching images taken at different times, or matching an image to a map (GDB). Registration of images to maps is viewed here as an important step in unlocking rich a priori knowledge stored in the data base for use in image analysis. To succeed, registration clearly must be achievable with only a partial primitive image analysis. Once registration is accomplished, more complete image analysis can be performed under map guidance and hopefully will be much more reliable and efficient than image analysis without a priori information.

Reasonable human and automatic registration procedures are based on "pass points". Pass points are uniquely identifiable points in the image usually defined by special edge context, i.e. points where rivers or roads intersect, corners of buildings, mountain peaks, etc. Humans will uniquely identify pass points because ambiguity can be removed by their large amount of global knowledge. Automatic procedures, on the other hand, typically work with far less global knowledge and must be built to tolerate ambiguities in matching single pass points from the image with those of the map. The next section gives a brief survey of some registration work which is used to set the stage for introducing the L.N.K. registration procedure developed in Section 3.2.

### 3.1 Survey of previous registration work

Mathematically, we formulate the registration problem as the problem of determining a transformation  $T_\alpha$  that maps a point  $P_1 = (x, y)$  in the first image space to "corresponding" point  $P_2 = (u, v)$  in the second image space. A definition of "corresponding" can be subtle. Points can correspond because of their obvious and unambiguous structure and/or interpretation, such as the tip of the Washington Monument as seen in two pictures. Points can also correspond by the relationship which  $T_\alpha$  places on them; -- i.e. a point  $(x, y)$  on the Mall lawn, in image one, corresponds to a point  $(u, v)$  in image two because,  $T_\alpha(x, y) = (u, v)$ . The first type of point will be called a pass point or control point. Pass points can be used to construct  $T_\alpha$  which can then be used to relate all other points in the image domain regardless of any obvious structural correspondence between the points. In general, we are less certain about the correspondence of points on the Mall lawn than we are about the Monument tips.

A straightforward and common registration technique is to use human selection of pass points, in the two images, resulting in a set of corresponding points  $C = \{(P_{11}, P_{21}), (P_{12}, P_{22}), \dots, (P_{1k}, P_{2k})\}$ . The desired transformation  $T_\alpha$  may have from two to six parameters depending on the particular model of image formation. The best transformation  $T_\alpha$  can be defined as that  $T_\alpha$ , such that

$$e(\alpha) = \sum_{i=1, k} d^2(T_\alpha(P_{1i}), P_{2i})$$

is minimized, where  $d^2$  is the distance between the control point  $P_{2i}$

in the second image, and the transform of the corresponding point  $P_{1i}$  from the first image. Classical least squares procedures can be used to determine  $T_\alpha$  [Van Wie 1977]. To automate the registration procedure a method of automatic selection of the points in SetC must be devised. The least squares fit of  $T_\alpha$  will be sensitive to any errors in this set of corresponding points.

Horn and Bachman [1977] have a procedure which uses all points in the image overlap and hence requires no feature selection to get Set C. However, the computation of  $e(\alpha)$  becomes involved and hill-climbing from an approximate  $T_\alpha$  must be used. Sometimes such an approximation is available from knowledge of the attitude of the sensing platform.

Barrow et al [1977] proposed using only salient edge points for set C and described a computationally fast method of computing  $e(\alpha)$  called "chamfer matching". Their procedure also used hill-climbing from an approximate  $\alpha$  to get a general 6-parameter transformation  $T_\alpha$ , valid for even the modeling of oblique imagery.

The heart of the registration problem is the selection of the set of control points C. In map making this is usually a human operation. Here we are exploring automatic techniques. If an approximate  $T_\alpha$  is available it is possible to search a limited area of an image to find a point corresponding (in structure) to a point in a map (second image). Block correlation can be used so long as  $T_\alpha$  effectively removes rotation. Van Wie and Stein [1977] report some success at this for repeat ERTS coverage. However, even use of gradient images for block correlation was

not always reliable. Deficiencies in block correlation have also been pointed out by Crombie [1975]. L.N.K. believes in the following two principles which have tempered the research work reported below.

- (1) Correlation for control point matching should consider higher level structure in the neighborhood defining the control point. For instance, edges or lines should be detected and their points of intersection typed as "T's" or "X's" before correspondences are attempted.
- (2) Correspondences drawn from local matching criteria must be tempered via feedback from global registration knowledge. For instance, many crossroads will individually look alike, but a globally valid registration transformation can pair them unambiguously, using the information contained in the spatial distribution of the entire set.

L.N.K. has developed a registration procedure which integrates both local and global matching criteria and has been quite successful in accounting for rotations and translations on vertical photography. The next section gives the formal development of the procedure and experiments on real data are reported in Section 3.3.

### 3.2 L.N.K. Registration Procedure

A technique has been developed by L.N.K. Corporation for automatically registering vertical photography with maps or other vertical photography. This technique has been successfully demonstrated at L.N.K. and on the ROSA system at the Research Institute of USAETL. The current assumptions (limitations) are that the registration transformation be limited to an arbitrary rotation  $\theta$  and a translation  $(\Delta x, \Delta y)$  which is no larger than half of the image diameter. Due to relief displacement and distortion, the derived  $T_\alpha$  is only an approximate transformation. However,  $T_\alpha$  could be refined as will be shown. In addition, by first deriving a global approximation it is likely that a more robust and efficient procedure is obtained. A method of introducing local adjustments to  $T_\alpha$  for matching individual features is covered in Section 5.

#### 3.2.1 The General Procedure

Registration is viewed as a three step procedure. Here the terminology pertains to matching image structure to a map or model. However, in fact, the map or model may be replaced by an image of the same or different type as has been done in several experiments.

(Step G1: feature extraction and local matching)

Obtain corresponding structures in the image and map. Structures correspond when they have the same shape, size, color, etc. The correspondences are plausible from local evidence but may be ambiguous or incorrect. In fact, each image structure may be paired with one, none, or many map structures.

(Step G2: global interpretation via  $T_\alpha$ )

Determine transformation parameters  $\alpha = (a_1, a_2, \dots, a_n)$  such that  $T_\alpha$  maps at least some image structures ("i.e. points, lines, arcs, etc.) onto corresponding map structures.

(Step G3: global match evaluation)

Determine the global goodness of match induced by  $T_\alpha$  on the image structures and map structures. (One way to do this is to compute the mean squared distances between map structures and image structures transformed by  $T_\alpha$ .)

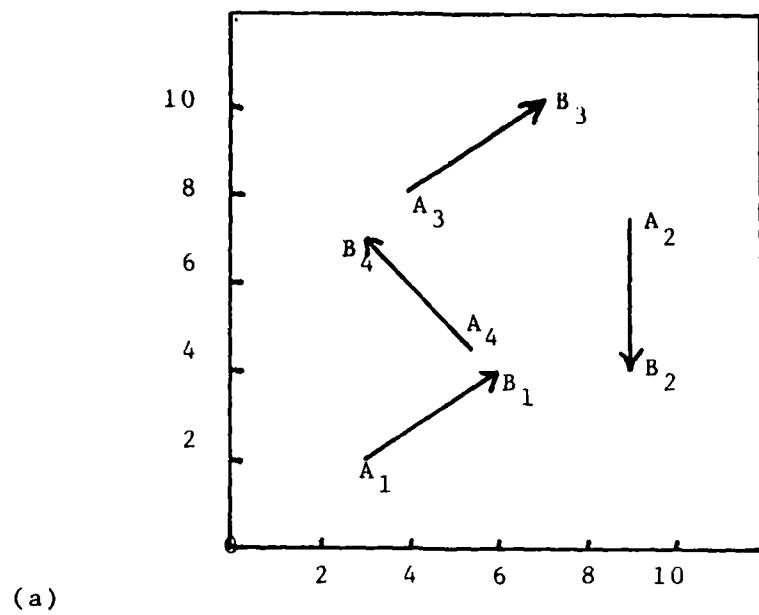
### 3.2.2 Specific registration procedure.

The specific L.N.K. registration technique is a hybrid of template matching and structural analysis and combines the advantages of those two procedures. The specific interpretation of the general steps above are as follows.

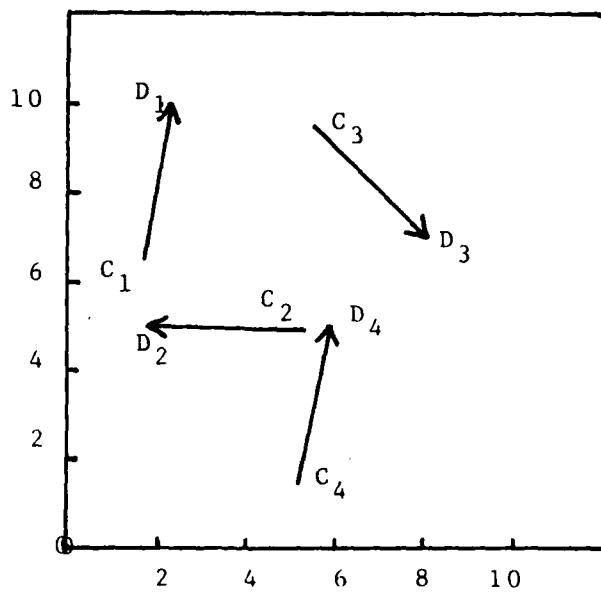
- (S1) Assume all structures of the same type correspond. For example, assume each straight line segment in the image can correspond to each straight line segment of the model, each convex curve in the image can correspond to each convex curve of the model, etc. For each pair of structures  $(s_i, s_m)$ , where  $s_i$  and  $s_m$  are structures from the image and map respectively, compute transformation parameters  $\alpha$  and place a unit of measure in  $\alpha$ - parameter space.
- (S2) Possible transformations  $T_\alpha$  between image and model are detected as clusters in  $\alpha$ - parameter space formed in step S1 because heavy measure at  $\alpha_0$  in  $\alpha$ - space means that many correspondences are explained by  $T_{\alpha_0}$ .
- (S3) Evaluation of the match strength of each  $T_\alpha$  from step S2 is obtained by either computing an average distance between all corresponding structures or by counting the number of image structures explained by the model structures under  $T_\alpha$ .

### 3.2.3 A Simple Example

A simple example of this process is illustrated in Figure 3.1. Assume that the image can be represented by the 4 directed edge elements shown in (a) while the map contains the edge elements in (b).



(a)



(b)

Figure 3.1 Example of global registration via clustering of local evidence. Image edge elements in (a) need to be rotated  $45^\circ$  and then translated (4.5, -2.) to be transformed into corresponding map edge elements in (b). (c) 16 units of measure are amassed in  $(\theta, xs, ys)$ -space forming a cluster at  $(\theta=0.79, xs=4.5, ys=-2.0)$ .

(c)

i	j	A	B	C	D	θ	xs	ys
1	1	3.0,2.0	6.0,4.0	1.7,6.4	2.3,10.0	0.82	1.1	2.8
1	2	3.0,2.0	6.0,4.0	5.3,5.0	1.8,5.0	2.55	8.9	5.0
1	3	3.0,2.0	6.0,4.0	5.5,9.5	8.0,7.0	4.91	3.0	12.1
1	4	3.0,2.0	6.0,4.0	5.1,1.5	5.8,5.0	0.79	4.4	-2.0
2	1	9.0,7.5	9.0,4.0	1.7,6.4	2.3,10.0	2.98	11.8	12.3
2	2	9.0,7.5	9.0,4.0	5.3,5.0	1.8,5.0	4.71	-2.2	14.0
2	3	9.0,7.5	9.0,4.0	5.5,9.5	8.0,7.0	0.79	4.4	-2.2
2	4	9.0,7.5	9.0,4.0	5.1,1.5	5.8,5.0	2.94	15.4	7.1
3	1	4.0,8.0	7.0,10.0	1.7,6.4	2.3,10.0	0.82	4.8	-2.0
3	2	4.0,8.0	7.0,10.0	5.3,5.0	1.8,5.0	2.55	13.1	9.4
3	3	4.0,8.0	7.0,10.0	5.5,9.5	8.0,7.0	4.91	-3.1	11.9
3	4	4.0,8.0	7.0,10.0	5.1,1.5	5.8,5.0	0.79	7.9	-7.0
4	1	5.5,4.5	3.0,7.0	1.7,6.4	2.3,10.0	5.33	-5.2	8.3
4	2	5.5,4.5	3.0,7.0	5.3,5.0	1.8,5.0	0.79	4.6	-2.1
4	3	5.5,4.5	3.0,7.0	5.5,9.5	8.0,7.0	3.14	11.0	14.0
4	4	5.5,4.5	3.0,7.0	5.1,1.5	5.8,5.0	5.30	-1.7	3.6

Figure 3.1 (continued)

It is assumed that the length of the edge elements is accurately known. There are 16 possible ways that an edge element from (a) can be paired with an edge element in (b). Each pairing yields unique transformation parameters  $(\theta, xs, ys)$  as shown in (c). Four of the 16 possible pairings yield a consistent interpretation -- rotate by  $\theta = 0.79$  radians and translate by  $(4.5, -2.0)$ . The parameters from the 4 correct pairings form a cluster in  $\alpha = (\theta, xs, ys)$  - space, while the parameters from incorrect pairings are sparsely distributed in the space. In practical cases there will be many more than 4 primitive structures and not all pairings will be possible (i.e. due to size or shape differences) so the presence of a cluster in the parameter space should be even more obvious.

### 3.2.4 A More Complex Example

The clustering phenomena observed above should always occur, as long as the vectors representing the image and model have many correct matches in common. Imperfect automatic feature detection or imperfect objects in the image would result in additional points in the cluster space or fewer points in the cluster space, but the character of the clusters should remain the same despite a fair amount of variation in the image features. This has been confirmed in experiments registering images to maps or other images.

Some features which are extracted from images automatically include straight edge segments, corners, points of high curvature, and regularly shaped spots (i.e. circles or rectangles). Experiments reported in Section 3.3 show that the probability of detecting

individual features can be as low as 20% and recognition of the whole can still be achieved using the proposed clustering technique.

Consider the features of terrain observed from a high altitude above Harrisburg, Pennsylvania (Figure 3.2). Five points of high curvature and six points of intersection of lineal features are identified in Figure 3.2b. For instance, point  $I_{13}$  is the junction between Route 15 and the Pennsylvania Turnpike, while point  $C_{12}$  is a point on Sherman Creek. The 30 vectors shown in Figure 3.2b are formed with a tip on an intersection point and tail on a point of high curvature. While the vectors of Figure 3.2b look nothing like the region they represent, they are very useful for registration. Figure 3.2c shows 10 vectors formed by features detected in an image taken from a perspective  $145^\circ$  off of the original perspective in Figure 3.2a and 3.2b. The feature points were selected independently by two researchers with the same instructions. There are  $10 \times 30 = 300$  possible matches of vectors from 3.2b with vectors from 3.2c, almost all of which are ignored because of differences in length. The result of automatic clustering in the transformation parameter space is shown in Figure 3.2d. The correct transformation parameters  $(\theta, \Delta x, \Delta y) = (145^\circ, 38, 115)$  are obtained first from the strongest cluster and there is little competition from other clusters. Under the best transformation all 10 image vectors are explained by the set of model vectors. Within the tolerance allowed, 11 of the 30 model vectors are "seen in the image". A root-mean-square match weight of 33.16% can be assigned to the match based on an average over 30 individual matches.

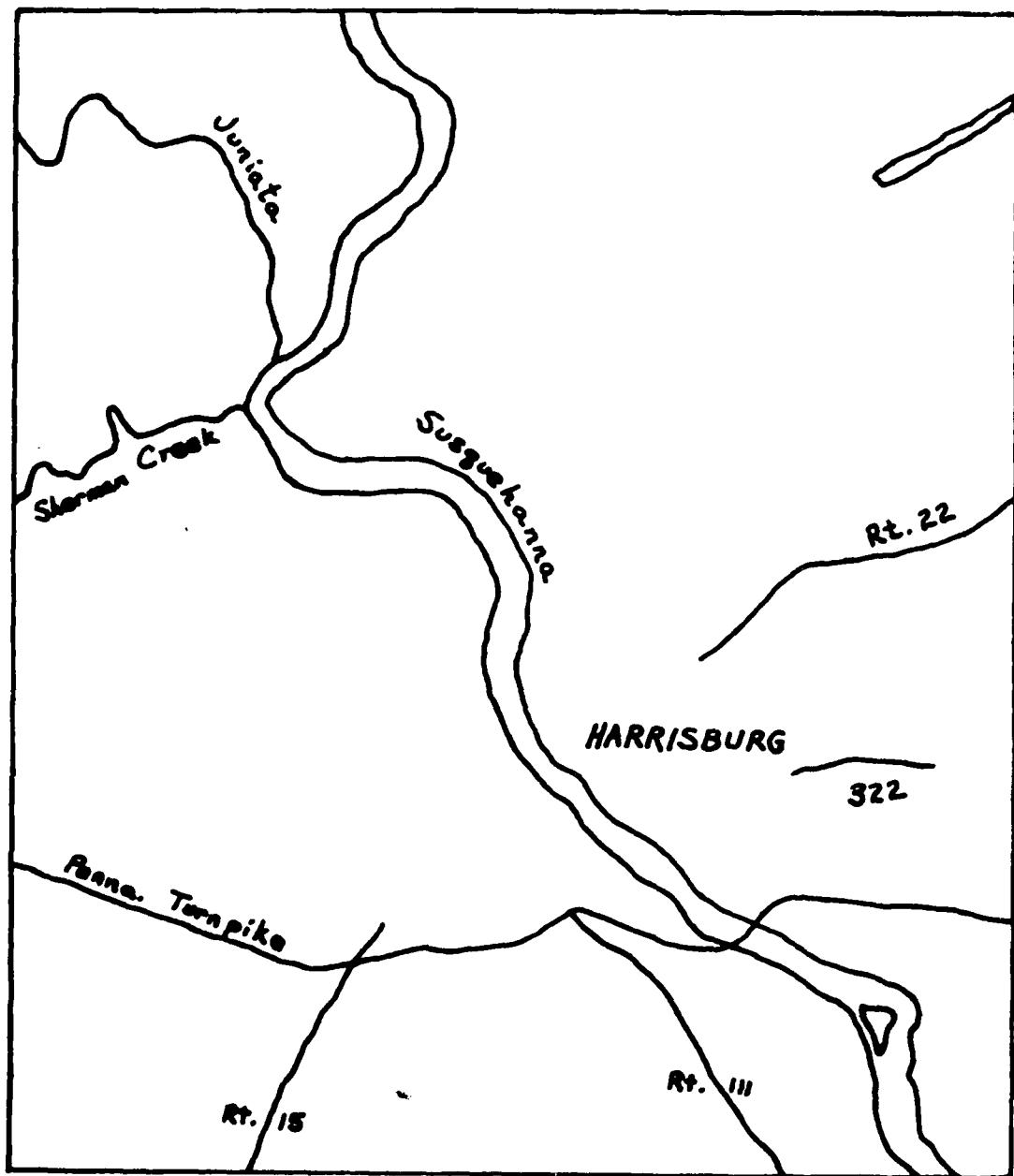


Figure 32 (a) Features of Harrisburg, Pennsylvania region which should be easily detectable in aerial imagery.

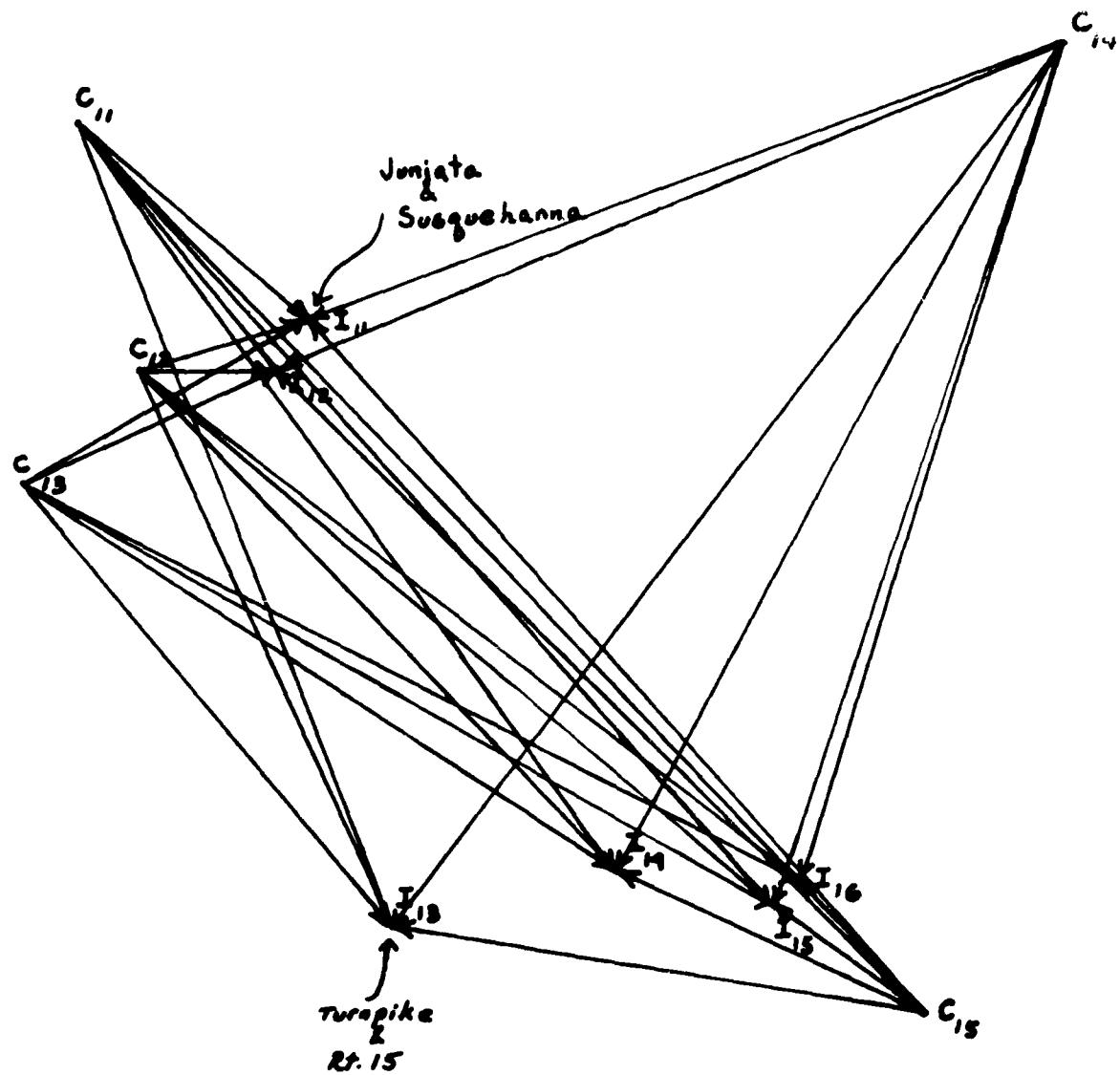


Figure 32(b) 5 points of curvature and 6 points of intersection detected by researcher #1 forming 30 abstract edges representing the scene in (a).

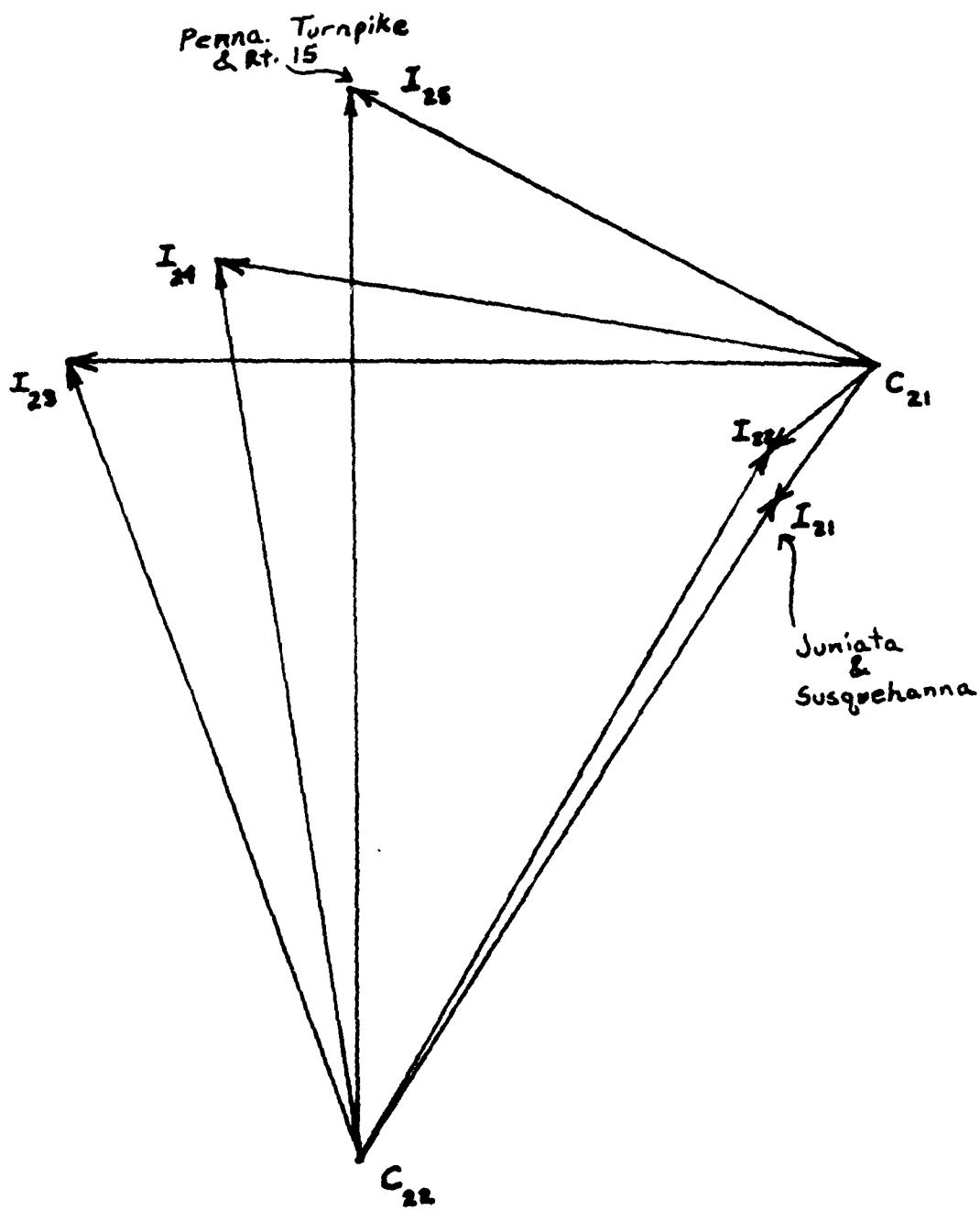


Figure 32(c) 2 points of curvature and 5 points of intersection detected by researcher #2 forming 10 abstract edges representing the scene in (a). Note that the perspective is quite different from that in (b).

@xgt reg.reg

@add reg.run2

INPUT 5,30,10,3,0,0,0,0,132,1

ANGTOL,STOL,DTOL,NUMLEV,PRT:BUK,SBK,MUM, PWIDTH,LENCHK

5 30 10 3 0 0 0 132 1

FOR IMAGE: LOX,LOY,MIX,HIY,XCENTER,YCENTER: -300 -500 300 200 0 -150

I SCALET(I) SCALEX(I) SCALEY(I)

1 36.000 60.000 70.000

2 7.200 12.000 14.000

3 1.440 2.400 2.800

LOX,LOY,LOTBND(1) = -300 -350 0

FOR MAP : LOX,LOY,MIX,HIY,XCENTER,YCENTER: -320 -320 320 320 0 0

CLUSTR

THETA	XSHIFT	YSHIFT	MTCHWT	MCHROW/NIMAGE	NMCHCL/NMAP
145	38.	115.	3316	10/ 10	11/ 30 ***
145	43.	123.	1603	9/ 10	10/ 30 ***
145	46.	123.	1366	7/ 10	7/ 30 ***
THETA	XSHIFT	YSHIFT	MTCHWT	MCHROW/NIMAGE	NMCHCL/NMAP
154	-58.	104.	310	1/ 10	1/ 30 ***
154	-53.	101.	316	1/ 10	1/ 30 ***
154	-50.	101.	320	1/ 10	1/ 30 ***
THETA	XSHIFT	YSHIFT	MTCHWT	MCHROW/NIMAGE	NMCHCL/NMAP
154	-46.	98.	323	1/ 10	1/ 30 ***
154	-43.	95.	323	1/ 10	1/ 30 ***
154	-41.	95.	326	1/ 10	1/ 30 ***
THETA	XSHIFT	YSHIFT	MTCHWT	MCHROW/NIMAGE	NMCHCL/NMAP
265	67.	-3.	856	3/ 10	3/ 30 ***
265	70.	-6.	853	3/ 10	3/ 30 ***
265	72.	-6.	886	3/ 10	3/ 30 ***
THETA	XSHIFT	YSHIFT	MTCHWT	MCHROW/NIMAGE	NMCHCL/NMAP
273	26.	-28.	293	1/ 10	1/ 30 ***
273	29.	-28.	280	1/ 10	1/ 30 ***
273	31.	-31.	290	1/ 10	1/ 30 ***
THETA	XSHIFT	YSHIFT	MTCHWT	MCHROW/NIMAGE	NMCHCL/NMAP
274	26.	-28.	326	1/ 10	1/ 30 ***
274	29.	-28.	323	1/ 10	1/ 30 ***
274	31.	-31.	323	1/ 10	1/ 30 ***
THETA	XSHIFT	YSHIFT	MTCHWT	MCHROW/NIMAGE	NMCHCL/NMAP
274	26.	3.	856	3/ 10	3/ 30 ***
274	29.	3.	853	3/ 10	3/ 30 ***
274	31.	3.	886	3/ 10	3/ 30 ***
THETA	XSHIFT	YSHIFT	MTCHWT	MCHROW/NIMAGE	NMCHCL/NMAP
274	26.	-28.	293	1/ 10	1/ 30 ***
274	29.	-28.	280	1/ 10	1/ 30 ***
274	31.	-31.	290	1/ 10	1/ 30 ***

### 3.2.5 Details of the L.N.K. registration procedure

Recall that the L.N.K. registration procedure consists of 3 conceptual steps. First, image structures and map structures must be paired so that for each possible corresponding pair, transformation parameters  $\alpha$  (or a set of several transformation parameters) can be computed which transform the image structure onto the map structure. Handling all such pairs populates the  $\alpha$ -space as discussed below. Second, clusters in  $\alpha$ -space must be detected so that those which cause a large amount of image/map overlay are discovered. Third, a quantitative measure of the amount of overlay is computed so that a transformation  $T_\alpha$  can be evaluated for acceptance, relative to some threshold criteria or relative to another possible  $T_\alpha$ .

Before proceeding it is essential to emphasize that the registration procedure is edge-based and can operate on either real or abstract edges. Real edges correspond to the location of real contrast changes as indicated in the gradient image. The direction of the edge indicates in which direction tonal increases occur. The length of a real edge in a map may be precise but it certainly is not precise in an edge element automatically extracted from imagery. Usually image edge elements represent some small segment of the map edge and in the experiments reported here are usually only 1/8 inch long. Abstract edges are formed by joining two arbitrary point features. The direction is arbitrary but fixed and length is fairly precise because the endpoints are detectable pass points. As shown in the experiments, although both kinds of edges

can be used, the abstract edges produce cleaner clustering and more precise transformations. The major reason is that pairs of corresponding abstract structures can be restricted to only those that agree very closely in length and exactly in end point type.

### 3.2.5.1 Pairing edge elements and populating $\alpha$ -space

We assume that  $\alpha$  specifies a rotation and translation:  $\alpha = (\theta, \Delta x, \Delta y)$ . If the orientation of the image edge element (EE) is  $\theta_i$  and the orientation of the map EE is  $\theta_m$  then it is easy to compute the rotational part of that  $\alpha$  that will overlay the two:  $\alpha = \theta_m - \theta_i$ . Referring to Figure 3.3 it is easily seen that  $\Delta x$  and  $\Delta y$  are then linearly constrained by the following equation.

$$\Delta x \cos \theta_m + \Delta y \sin \theta_m + (r_i - r_m) = 0$$

Actually the rotated EE A'B' is not free to lie anywhere along the line determined by points C and D but must lie along the segment C'D'. Thus the linear equation above can be replaced by two points between which  $(\Delta x, \Delta y)$  must lie. The first point is easily determined by that  $(\Delta x_1, \Delta y_1)$  necessary to translate point A' onto point C. Similarly the other extreme is determined by that  $(\Delta x_2, \Delta y_2)$  necessary to translate point B' onto point D. In practice, we allow for some overshoot (  $dtol$  ) of points C and D because the Hough detector may in fact overshoot the real edge somewhat when EE AB is detected and because of error in  $\theta_i$ . Errors of  $2^\circ$  in  $\theta_i$  are quite common for the 32x32 window used.

Thus for each pair of corresponding EE's the potential registration transformations are triples  $\alpha = (\theta, \Delta x, \Delta y)$  such that  $\theta = \theta_m - \theta_i \pm \theta_{\text{error}}$

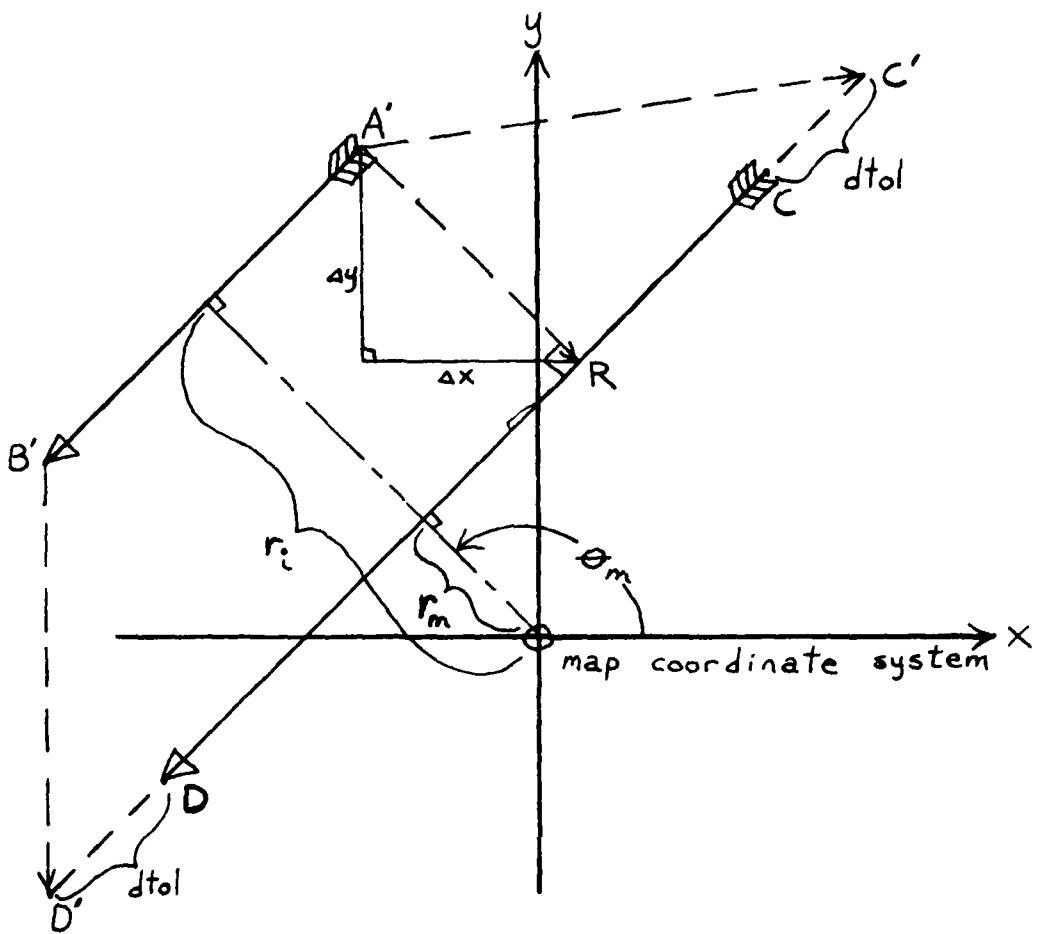


Image edge AB or  $(\theta_i, r_i)$  is rotated  $\theta_m - \theta_i$  into A'B' or  $(\theta_m, r_i)$  to be parallel to map edge CD or  $(\theta_m, r_m)$ . Unit vector from A' to R is  $(-\cos\theta_m, -\sin\theta_m)$ . Projection of A'C onto A'R has constant length for all C on line yielding following relation between  $\Delta x$  and  $\Delta y$ .

$$A'C \cdot (-\cos\theta_m, -\sin\theta_m) = r_i - r_m$$

$$(\Delta x, \Delta y) \cdot (-\cos\theta_m, -\sin\theta_m) = r_i - r_m$$

$$\Delta x \cos\theta_m + \Delta y \sin\theta_m + (r_i - r_m) = 0$$

More simply,  $(\Delta x, \Delta y)$  lies on a line segment between the two points  $(\Delta x_1, \Delta y_1) = (Cx', Cy') - (Ax', Ay')$  and  $(\Delta x_2, \Delta y_2) = (Dx', Dy') - (Bx', By')$  assuming that map edge CD is at least as long as image edge AB.

Figure 3.3 Derivation of constraints on  $\alpha = (\theta, \Delta x, \Delta y)$  for match of image edge element  $(\theta_i, r_i)$  and map edge element  $(\theta_m, r_m)$ .

and  $(\Delta x, \Delta y)$  lies between  $(\Delta x_1, \Delta y_1)$   $(\Delta x_2, \Delta y_2)$ . The error in  $\Delta x, \Delta y$  induced by  $\theta_{\text{error}}$  is of the order  $r\theta_{\text{error}}$ . By centering the origin for rotation in the window, the effect of this error is minimized. Assuming that  $r=2000$  stagels and that  $\theta_{\text{error}} < 2$  degrees, the error induced in  $(\Delta x, \Delta y)$  by  $\theta_{\text{error}}$  is at most  $2000 \text{ stagels} \times 2 \text{ degrees} / 360 \text{ degrees} \times 2\pi = 60$  stagels. Due to the threshold used for straight line detection, the Hough detector could overshoot the end of an edge by half of the window size or 128 stagels. By allowing another 60 stagels for error we get an approximation of 188 stagels for DTOL. Thus by theoretical considerations alone, we arrive at clustering parameters ANGTOL=5 degrees, STOL=50 stagels, and DTOL=188 stagels. Most of the experiments reported below were performed, before this theoretical analysis, with ANGTOL=5 degrees, STOL=50 stagels and DTOL=200 stagels yielding good results.

With  $\alpha$  so constrained there may be theoretically an infinite number of points to contribute to  $\alpha$ -space. However, the inherent error on  $\theta$ ,  $\Delta x$ , and  $\Delta y$  allows a discretization on  $\alpha$ -space which makes clustering by binning a good technique. For each pair of corresponding structures we place the 5-tuple  $(\theta, \Delta x_1, \Delta y_1, \Delta x_2, \Delta y_2)$  in our cluster space. We successively zoom in on clusters by redefining a grid of bins and for each level of clustering the set of 5-tuples is examined for incrementing the bins. Whenever points  $(\Delta x_1, \Delta y_1)$  and  $(\Delta x_2, \Delta y_2)$  are in separate bins, more than one bin count is incremented.

### 3.2.5.2 Clustering in $\alpha$ -space

In clustering by binning we work with a fixed number of bins. If the  $\theta$ ,  $\Delta x$ , and  $\Delta y$  dimensions are quantized into 10 units then there must be  $10 \times 10 \times 10$  bins. (Actually there will be more as explained later.) The "size" of each bin is determined by the range of the variable divided by 10. The range of the variable depends on the level of clustering. For instance, at the first level the range of  $\theta$  is  $360^\circ$  so each bin represents  $36^\circ$  along the  $\theta$  dimension. We always assume that  $\Delta x$  and  $\Delta y$  can vary no more than half of the diameter of the image so for a  $8000 \times 8000$  stagel image the range of  $\Delta x$  and  $\Delta y$  is 4000 and the size of each bin is then  $36^\circ \times 400$  stagels  $\times 400$  stagels at level one. When a level one bin is found to be "heavy" we can zoom in by resetting the full ranges of  $\theta$ ,  $\Delta x$ , and  $\Delta y$  to be those taken on for the heavy bin. Thus at level two the size of a bin is  $3.6^\circ \times 40$  stagels  $\times 40$  stagels and by level three it is  $0.36^\circ \times 4$  stagels  $\times 4$  stagels. Clearly three levels of clustering will be too much for typical problems. Clustering should proceed only until the bin size is comparable to the error inherent in determining  $\theta$ ,  $\Delta x$  and  $\Delta y$ . At any level, a detection is only made when an acceptable number of points lie in the bin. The acceptance threshold will be dependent upon the reliability of the edge or pass point detectors. If we demand that half of the detected edge elements should fall on edges existing in the map then our cluster detection threshold should be  $0.5N_e$  where  $N_e$  is the number of detected edge elements. In the experiments documented in Section 3.3 no absolute cluster threshold was applied. Instead, up to 3 of the best clusters at up to 3 levels

were pursued and the corresponding transformations  $T_\alpha$  were evaluated by a technique described in Section 3.2.5.3.

Clustering by binning can suffer severely from boundary effects as illustrated in Figure 3.4. A cluster actually present can be missed due to the quantization of the  $\alpha$ -space. To avoid the distribution of points from a cluster to several neighboring bins, 3 other overlapping or offset sets of bins can be used. The bins of the other grids are displaced by 1/2 of the bin size relative to the first grid of bins (See Figure 3.4). Suppose that a cluster of  $N$  points exists in  $\alpha$ -space and has diameter  $d$ . With the above technique there is guaranteed detection at threshold  $N$  so long as the bin size is at least  $2d$ . If bin size below  $2d$  is used, the cluster will be broken up and detection of  $T_\alpha$  will be jeopardized.  $d$  is determined from the error inherent in the parameters of the space. As previously stated, the error is not only due to measurement error but also contains approximation error since we are using only a rotation and translation whereas the true  $T_\alpha$  may have nonlinear effects. In the case of registration of stereo images differential displacements will occur which will be a function of elevation off some base level. The component of error due to elevation is bounded by considering the maximum relief to be observed in the imagery.

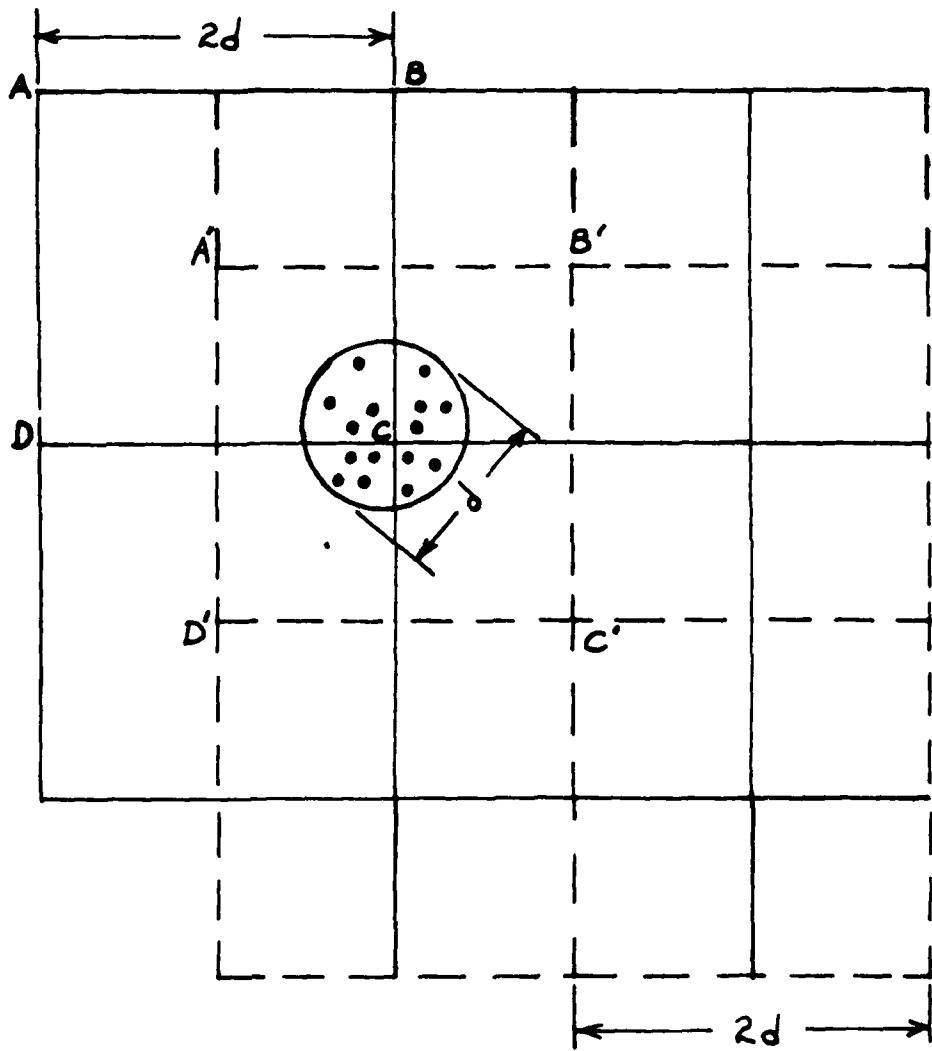
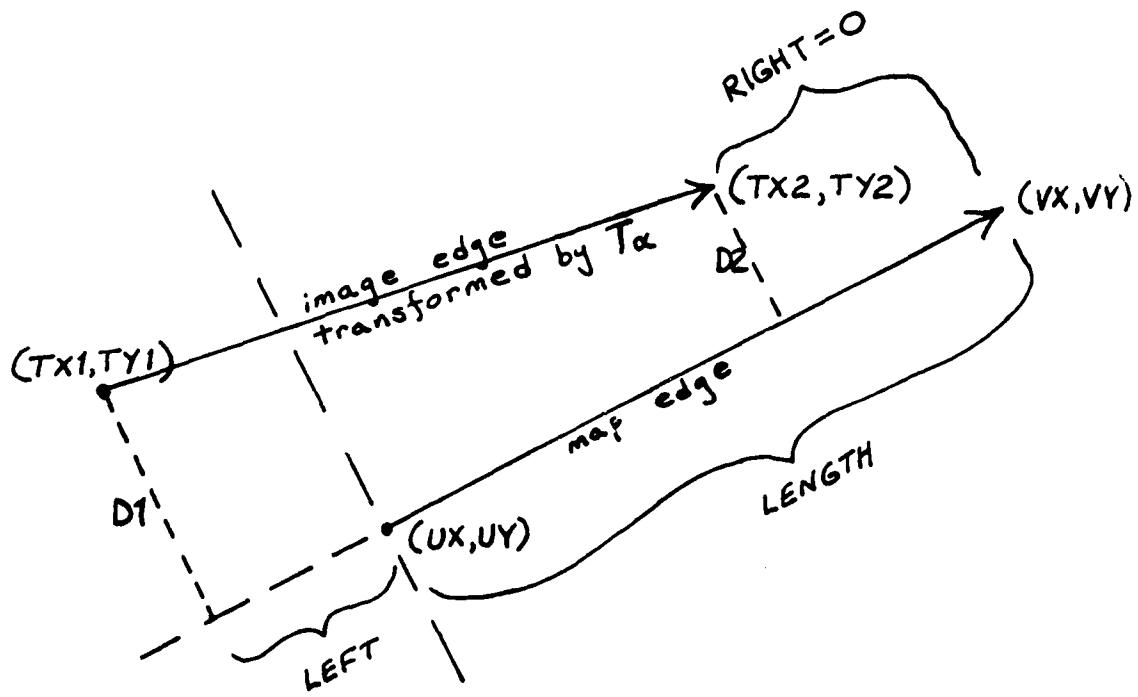


Figure 3.4 Clustering using 4 sets of overlapping bins.  
 No bin in first grid contains more than 4  
 points but 15 points enter bin ABCD in one  
 offset grid.

### 3.2.5.3 Evaluating the goodness of fit for $T_\alpha$

For matching point patterns the root-mean-square (RMS) distance between map points and transformed image points is a good measure of the match. For abstract edges the RMS technique is also applicable since two precise points determine each abstract edge. However, for real edges the point concept is not useful since it is unlikely that the whole of the image edge is present. A heuristic RMS distance computation is described below which was used to evaluate each candidate  $T_\alpha$  in the reported registration experiments.

Given a candidate  $T_\alpha$  determined by  $\alpha = (\theta, \Delta x, \Delta y)$  we can heuristically evaluate how good image edge  $(x_1, y_1) - (x_2, y_2)$  overlays map edge  $(u_x, u_y) - (v_x, v_y)$  under  $T_\alpha$ . Refer to Figure 3.5. A function is constructed which measures how well  $T_\alpha$  aligns the edges in the map space in terms of direction and proximity. Let  $(T_x_1, T_y_1) - (T_x_2, T_y_2)$  be the image edge transformed under  $T_\alpha$ . If this transformed edge has direction which differs from the map edge  $(u_x, u_y) - (v_x, v_y)$  by more than the angle tolerance ANGTOL then the match weight is 0 ( $EMATCH(T_\alpha, (x_1, y_1)(x_2, y_2), (u_x, u_y)(v_x, v_y)) = 0$ ). ANGTOL should be set according to the  $\theta$  error in  $T_\alpha$ . Most of the possible pairings of image and map edges will have EMATCH of 0 by failing this test. If the transformed image edge has direction compatible with the map edge then the EMATCH value is computed as the product of a length overlap and a proximity value. To get the length overlap LMATCH, the transformed edge is projected onto the map edge and any LEFT or RIGHT overshoots are measured and adversely affect the LMATCH as shown in Figure 3.5. Note that the LMATCH can



$$LMATCH = \max \left\{ 0, \frac{LENGTH - LEFT - RIGHT}{LENGTH} \right\}$$

$$DMATCH = \max \left\{ 0, 1 - \frac{D1^2 + D2^2}{2 * DTOL^2} \right\}$$

$$EMATCH = LMATCH * DMATCH$$

Figure 3.5 Computation of a match score between a map edge and an image edge under transformation  $T_\alpha$ .

be 1.0 even if the image edge is much shorter than the map edge. A proximity value or DMATCH is computed by comparing the average distance between points on the edges and a distance tolerance DTOL. DTOL should be set from the error inherent in  $\Delta x$  and  $\Delta y$ .

Edge pairs which have EMATCH = 0 are assumed not to correspond to the same image structure while those pairs with EMATCH  $\neq$  0 are assumed to be corresponding. Theoretically each abstract or real image edge should correspond with at most one map edge while each real map edge may correspond to many real edges (short segments of the same edge). The L.N.K. registration software records all correspondences made by using the EMATCH function on all possible edge pairs. A "match matrix" may be optionally output when  $T_\alpha$  is evaluated. Not only does the match matrix show the goodness of  $T_\alpha$  but it also shows where verification must be done to locate missing image edges or where positive change might have been detected due to image edges without matches in the map.

### 3.3 Registration Experiments

This section describes some experiments on registering aerial imagery. In each case the "true" transformation is provided, for comparison purposes. This transformation was obtained by calculating the transformations that aligned known corresponding abstract edges. Due to distortions present, such as those caused by the relief of the terrain, this "true" transformation is only approximate and ranges on the variability have been estimated and provided. This fact should be taken into account when evaluating the registration results.

#### 3.3.1 4621 Image

The first set of experiments used the image designated "4621" and shown in Figure 1.1. In order to determine the "true" transformation, registration was performed using abstract edges from the image and its map. The abstract edges were formed by connecting intersection points in the following way:

- (1) L-intersection points to Y- or T-intersection points,
- (2) L-intersection points to X-intersection points,
- (3) Y- or T-intersection points to X-intersection points.

The intersection points for the image were obtained using an interactive sampling program under human guidance. The abstract edges were then constructed by hand, and the result is shown in Figure 3.6.

A map of 4621 was created by using a Talos digitizer to extract a set of pass points. Most of these pass points were intersection points and the entire set is listed in Table 3.1. A set of abstract edges was constructed using the same criteria used for the image abstract edges, and the result is presented in Figure 3-7.

The registration procedure was performed on the two sets of abstract edges and the result appears in Figure 3.8. 30 of the 43 image abstract edges

Table 3-1      Digitization of points on image 4621 using Talos digitizer (0.001 inch resolution) and scanning stage (0.0005 inch resolution).

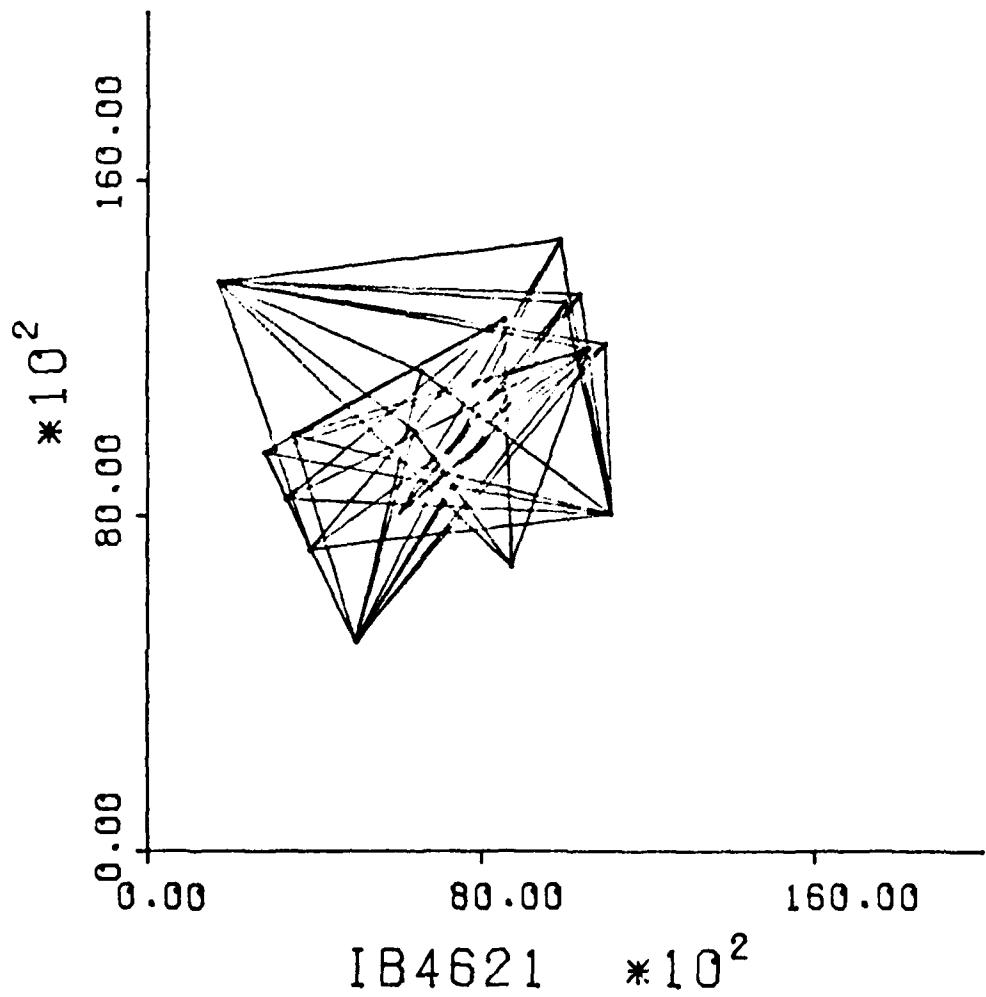
IMAGE PT # (see Fig. 1.1)	TALOS *		STAGE *		DESCRIPTION **
	X	Y	X	Y	
1	2498	3641	5000	5000	X
2	4365	2747	8722	6803	T
3	5583	2191	11120	8052	X
4	3044	4744	3913	7203	T
5	3342	5357	3314	8405	T
6	3611	5908	2814	9511	T
7	1980	6678	-	-	T
8	3975	5750	3508	9946	T
9	5516	5073	6563	11448	T
10	7130	3514	10964	12108	Y
11	7430	4091	10367	13304	L
12	7723	4700	9866	14600	L
13	8642	4258	-	-	X
14	6952	3223	11209	11461	A
15	6942	3613	10565	12000	L
16	7233	4181	10010	13101	T
17	4861	7575	1710	13600	X
18	5980	7667	-	-	Y
19	6444	7480	-	-	T
20	9471	6190	-	-	T
21	5013	7886	1399	14295	A
22	4480	7729	-	-	T
23	7346	9632	-	-	A
24	9282	5763	-	-	A
25	9700	6104	-	-	Y
26	5279	7999	1513	14804	A
27	4010	1991	9388	5480	A
28	5480	2552	-	-	L
29	4808	2527	-	-	A

\* The photo in Figure 1.1 was rotated roughly 50 degrees clockwise when mounted on the digitizer. The corresponding transparency was not rotated when mounted on the stage.

\*\* L,X,T, and Y indicate type of intersection, while A indicates an arbitrary point on a straight road segment.

43,1000,3000,14000,16000,	10010,13101,5000,5000,	6402,10030,8722,6803,
10565,12000,3508,9946,	10010,13101,11120,8052,	10565,12000,6050,8068,
10565,12000,2814,9511,	10964,12108,1710,13600,	6402,10030,1710,13600,
10565,12000,3314,8405,	10964,12108,5000,5000,	6402,10030,5000,5000,
10565,12000,3913,7203,	10964,12108,11120,8052,	9866,14600,6050,8068,
10565,12000,8722,6803,	6563,11448,1710,13600,	10367,13304,6050,8068,
10565,12000,1710,13600,	6563,11448,5000,5000,	10010,13101,6050,8068,
10565,12000,5000,5000,	6563,11448,11120,8052,	10964,12108,6050,8068,
9866,14600,1710,13600,	3508,9946,5000,5000,	3508,9946,8532,12698,
9866,14600,5000,5000,	3508,9946,11120,8052,	2814,9511,8532,12698,
9866,14600,11120,8052,	2814,9511,5000,5000,	3314,8405,8532,12698,
10367,13304,1710,13600,	2814,9511,11120,8052,	3913,7203,8532,12698,
10367,13304,5000,5000,	3314,8405,11120,8052,	8722,6803,8532,12698,
10367,13304,11120,8052,	3913,7203,1710,13600,	
10010,13101,1710,13600,	3913,7203,11120,8052,	
	8722,6803,1710,13600,	

(a)



(b)

Figure 3.6 Listing (a) and plot (b) of edges from 4621

AD-A101 319

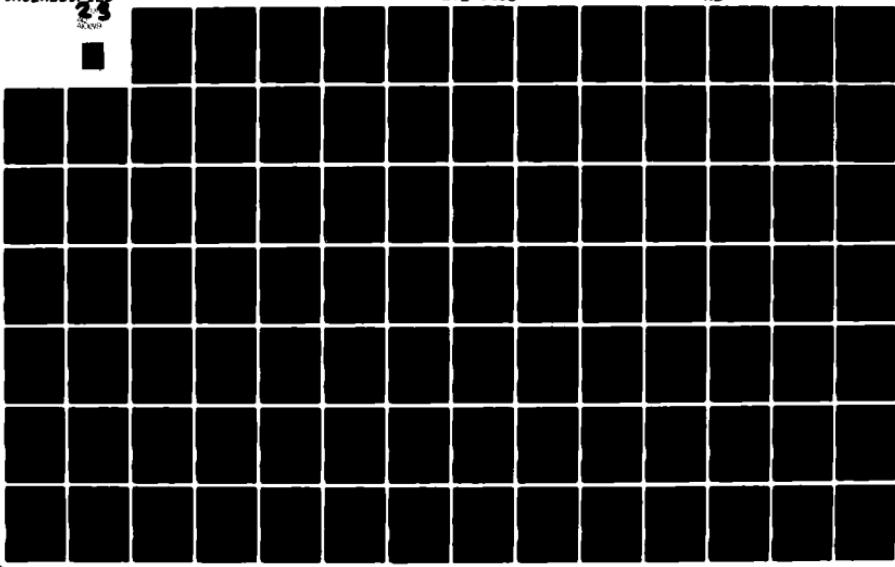
L N K CORP SILVER SPRING MD  
KNOWLEDGE-BASED IMAGE ANALYSIS. (U)  
APR 81 G C STOCKMAN, B A LAMBIRD, D LAVINE  
ETL-0258

F/6 9/2

DAAK70-77-C-0110  
NL

UNCLASSIFIED

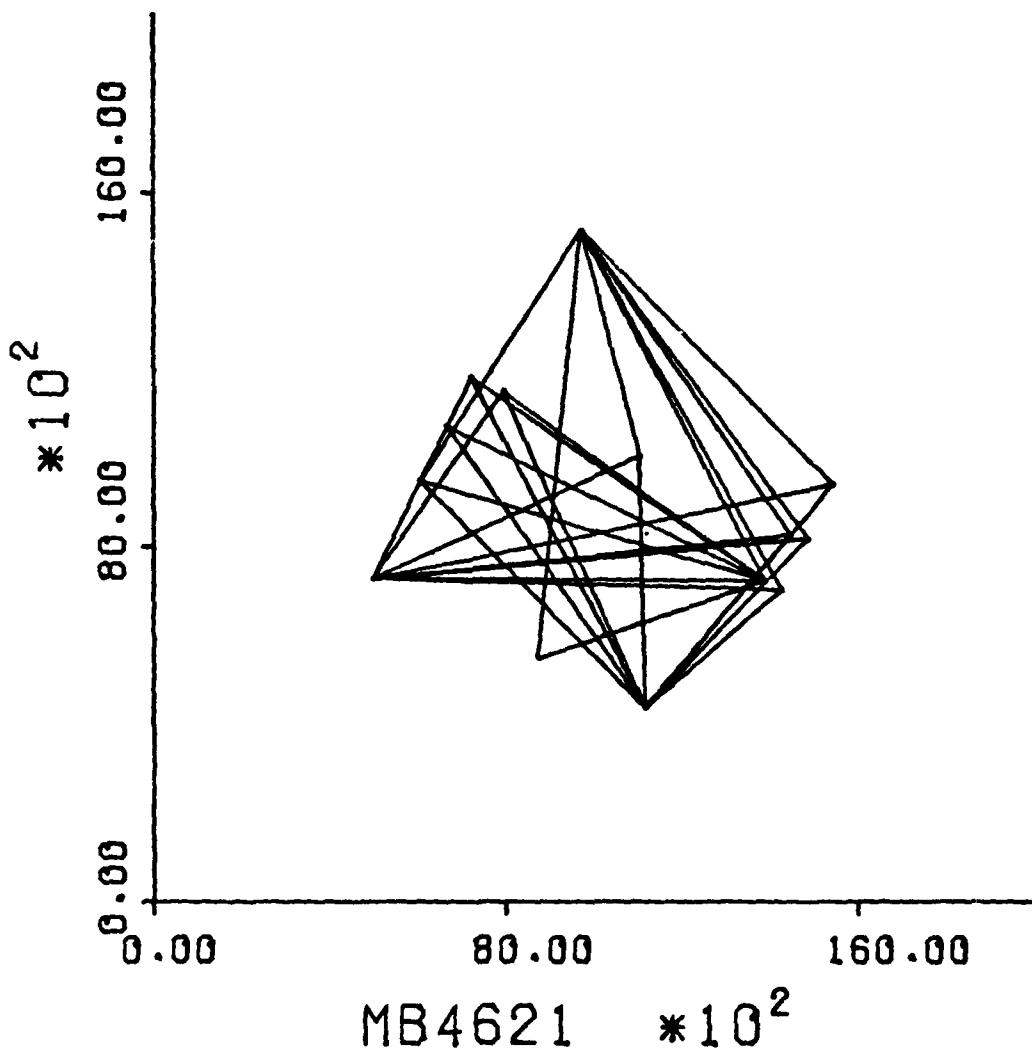
23  
4000



30,3000,3000,16000,16000,  
 13884,7226,7950,11500,  
 13884,7226,7222,11816,  
 13884,7226,6684,10714,  
 13884,7226,6088,9488,  
 13884,7226,8730,5494,  
 13884,7226,9722,15150,  
 13884,7226,4996,7282,  
 15446,9400,9722,15150,  
 15446,9400,4996,7282,  
 15446,9400,11166,4388,  
 14860,8182,9722,15150,  
 14860,8182,4996,7282,  
 14860,8182,11166,4388,  
 14466,8262,9722,15150,  
 14466,8262,4996,7282,

14466,8262,11166,4388,  
 14260,7028,9722,15150,  
 14260,7028,4996,7282,  
 14260,7028,11166,4388,  
 11032,10046,9722,15150,  
 11032,10046,4996,7282,  
 11032,10046,11166,4388,  
 7950,11500,4996,7282,  
 7950,11500,11166,4388,  
 7222,11816,4996,7282,  
 7222,11816,11166,4388,  
 6684,10714,11166,4388,  
 6088,9488,9722,15150,  
 6088,9488,11166,4388,  
 8730,5494,9722,15150,

(a)



(b)

Figure 3.7 Listing (a) and plot (b) of abstract map edges from 4621

```

INPUT 5,50,200,3,0,0,0,132,1,20,
ANGTOL,STOL,DTOL,NUMLEV,PRT:SBK,SBK,MWM, PWIDTH,LENCHK,THREBH =
5 50 200 3 0 0 0 132 1 20
GOING TO RDCNTR, IONT, DONT= 1 6
FOR IMAGE: LOX,LOY,HIX,HIY,XCENTER,YCENTER: 1000 30001400016000 7500 9500
0 1 SCALEx(i) SCALEx(i) SCALEx(i)
1 36.808 1300.000 1300.000
2 7.200 260.000 260.000
3 1.440 52.000 52.000
LOX,LOY,LOTBND(1) = -6500 -6500 0
FOR MAP : LOX,LOY,HIX,HIY,XCENTER,YCENTER: 3000 30001600016000 9500 9500
CLUSTR
0 NIIMAGE,NMAP= 43 30
0THETA XSHIFT YSHIFT HTCHWT HCHROW/NIIMAGE NMCHCL/NMAP
309 598. -1378. 8610 30/ 43 30/ 30 88888
309 624. -1352. 8430 30/ 43 30/ 30 88888
309 572. -1352. 8633 30/ 43 30/ 30 88888
0THETA XSHIFT YSHIFT HTCHWT HCHROW/NIIMAGE NMCHCL/NMAP
309 598. -1378. 8610 30/ 43 30/ 30 88888
309 624. -1352. 8430 30/ 43 30/ 30 88888
309 572. -1352. 8633 30/ 43 30/ 30 88888
0THETA XSHIFT YSHIFT HTCHWT HCHROW/NIIMAGE NMCHCL/NMAP
309 598. -1378. 8610 30/ 43 30/ 30 88888
309 624. -1352. 8430 30/ 43 30/ 30 88888
309 572. -1352. 8633 30/ 43 30/ 30 88888
*** END OF CLUSTERING FOR THIS DATA SET ***

```

Figure 3.8 Results obtained from registration software when run with set of abstract map edges and the 100% reliable set of abstract image edges from 4621

were matched correctly with all 30 of the map abstract edges, giving the "true" transformation ( $\Delta\theta=309^\circ \pm 1^\circ$ ,  $\Delta x=598 \pm 60$ ,  $\Delta y=-1378 \pm 40$ ).

### 3.3.1.1 Registration using real edges.

The image edges were found using the lineal feature detection process discussed in Section 2. A set of 104 real edges were found and are shown in Figure 2.9a. The map edges were constructed by connecting intersection points listed in Table 3-1 that were actually connected in the image. This set is presented in Figure 3.9.

The results of registering the real edges appear in Figure 3.10. The transformation with the highest weight was  $(307^\circ, 539, -1371)$  which matched 63 of the 104 image edges with 9 of the 18 map edges. The difference in the transformation parameters from the "true" values would lead to a difference in position of 77 stagels or .039 inches on the image for a point at  $(1000, 1000)$ .

### 3.3.1.2 Robustness of Registration Procedure

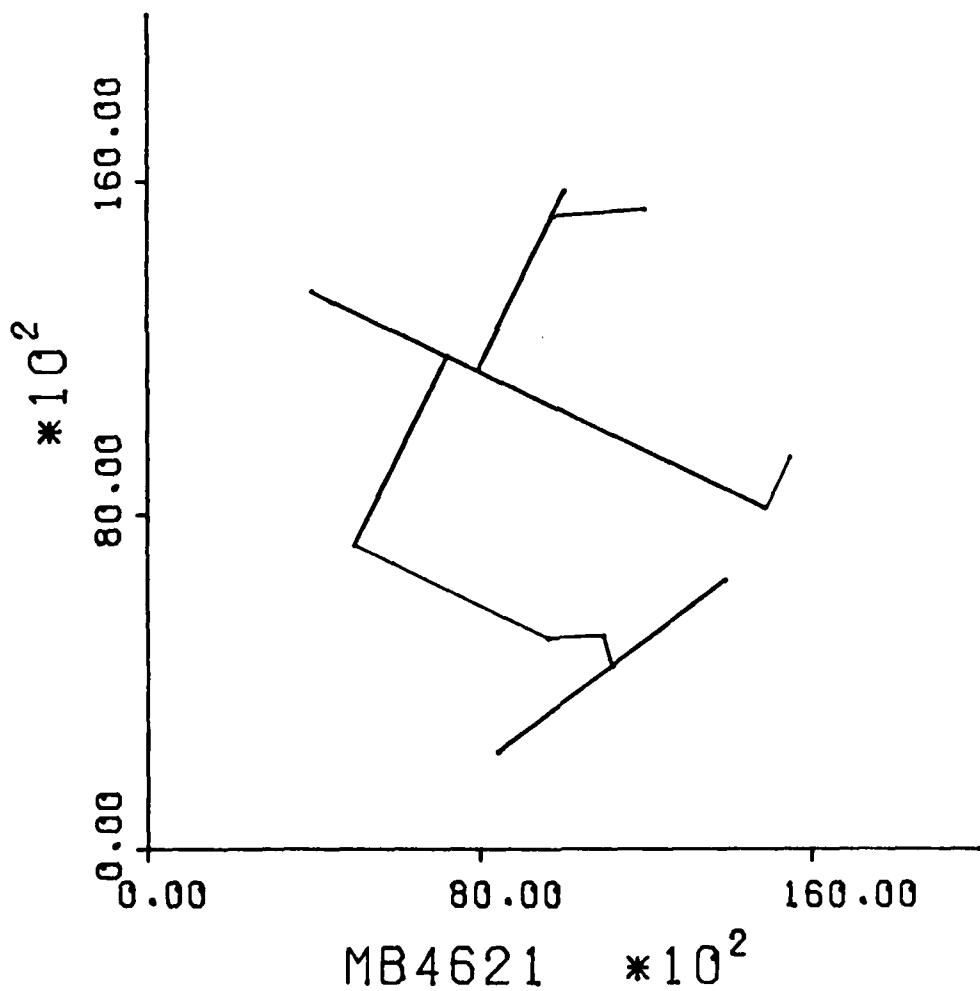
The experiment described above assumed a 100% reliable feature detector since the features were chosen under human control. Since the likelihood of obtaining this degree of reliability with a completely automatic feature detector is low, experiments were performed on abstract edge sets with varying degrees of reliability. The detection probabilities used were 80%, 60%, 40%, and 20%. Thus, feature points were chosen with a probability of 80%, or 60%, etc. and abstract edges were formed using the same criteria as before.

The 80%-reliable edge set was created by using a table of random numbers to set up a selection process such that each feature point had an 80% probability of detection. The selection process used was to look at a sequence of

18,3000,3000,16000,16000,  
4996,7282,9616,5054,  
9616,5054,4996,7282,  
4996,7282,7222,11816,  
7222,11816,4996,7282,  
3960,13356,14860,8182,  
14860,8182,3960,13356,  
7950,11500,10026,15772,  
10026,15772,7950,11500,  
9722,15150,11960,15334,

11960,15334,9722,15150,  
14860,8182,15446,9400,  
15446,9400,14860,8182,  
10960,5108,11166,4388,  
11166,4388,10960,5108,  
10960,5108,9616,5054,  
9616,5054,10960,5108,  
8428,2330,13904,6446,  
13904,6446,8428,2330,

(a)



(b)

Figure 3.9 Listing (a) and plot (b) of real map edges from 4621

INPUT 5,50,200,3,0,0,0,132,0,0,  
 INPUT 5,50,200,3,0,0,0,132,0,0,  
 ANGTOL,STOL,DTOL,NUMLEV,PRT:BUK,SBK,MWM, PWIDTH,LENCHK,THRESH =  
 5 50 200 3 0 0 0 132 0 0  
 FOR IMAGE: LOX,LOY,HIX,HIY,XCENTER,YCENTER: 1000 30001400016000 7500 9500  
 0 I SCALET(I) SCALEX(I) SCALEY(I)  
 1 36.000 1300.000 1300.000  
 2 3.600 130.000 130.000  
 3 .360 13.000 13.000  
 LOX,LOY,LOTBND(1) = -6500 -6500 0  
 FOR MAP : LOX,LOY,HIX,HIY,XCENTER,YCENTER: 3000 30001600016000 9500 9500  
 CLUSTR  
 CLUSTR  
 0 NIMAGE,NMAP= 104 18  
 0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP  
 307 832. -1248. 2172 45/ 104 6/ 18 \*\*\*\*  
 307 832. -1235. 2105 44/ 104 6/ 18 \*\*\*\*  
 307 767. -1287. 2966 50/ 104 8/ 18 \*\*\*\*  
 0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP  
 307 767. -1287. 2966 50/ 104 8/ 18 \*\*\*\*  
 307 780. -1274. 2738 48/ 104 7/ 18 \*\*\*\*  
 307 715. -1248. 3327 52/ 104 9/ 18 \*\*\*\*  
 0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP  
 307 858. -1300. 2044 46/ 104 7/ 18 \*\*\*\*  
 307 916. -1241. 1505 41/ 104 4/ 18 \*\*\*\*  
 307 929. -1228. 1427 41/ 104 4/ 18 \*\*\*\*  
 0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP  
 307 546. -1365. 4822 63/ 104 9/ 18 \*\*\*\*  
 307 539. -1371. 4827 63/ 104 9/ 18 \*\*\*\*  
 307 539. -1358. 4822 63/ 104 9/ 18 \*\*\*\*  
 0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP  
 307 468. -1326. 4405 63/ 104 9/ 18 \*\*\*\*  
 307 494. -1313. 4500 63/ 104 9/ 18 \*\*\*\*  
 307 474. -1319. 4400 63/ 104 9/ 18 \*\*\*\*  
 0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP  
 307 338. -1404. 3505 59/ 104 7/ 18 \*\*\*\*  
 307 351. -1391. 3555 58/ 104 7/ 18 \*\*\*\*  
 307 377. -1378. 3738 60/ 104 8/ 18 \*\*\*\*  
 0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP  
 307 650. -1170. 3244 47/ 104 7/ 18 \*\*\*\*  
 307 676. -1157. 3094 47/ 104 7/ 18 \*\*\*\*  
 307 702. -1144. 3027 46/ 104 6/ 18 \*\*\*\*  
 0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP  
 307 585. -1209. 3761 52/ 104 9/ 18 \*\*\*\*  
 307 559. -1222. 3888 55/ 104 9/ 18 \*\*\*\*  
 307 572. -1209. 3738 51/ 104 9/ 18 \*\*\*\*  
 0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP  
 307 611. -1261. 4311 59/ 104 9/ 18 \*\*\*\*  
 307 585. -1287. 4544 62/ 104 9/ 18 \*\*\*\*  
 307 598. -1274. 4444 61/ 104 9/ 18 \*\*\*\*  
 \*\*\* END OF CLUSTERING FOR THIS DATA SET \*\*\*

Figure 3.10 Results obtained from registration software when run  
 with set of real map edges and real image edges from 4621,  
 EMATCH = MTCHWT / 10000 = .48.

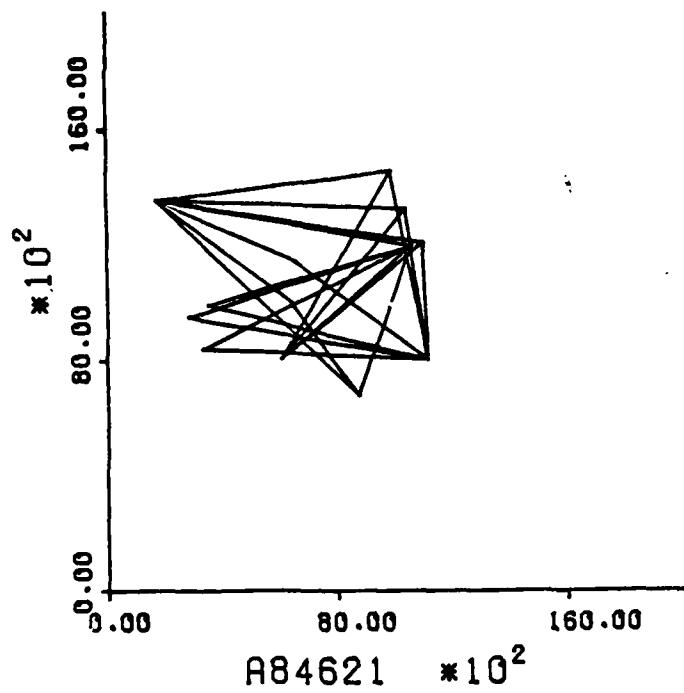
random numbers and eliminate those numbers ending in a "1" or a "2". The remaining sequence numbers were then used to select the feature points from the original set. The 60% and other sets were constructed similarly. Note that no false point detections were added to any of the sets.

The resulting edge sets are shown in Figure 3-11. Two 40%-reliable sets were created and three 20%-reliable sets were created. The registration results are presented in Table 3-2. For each experiment the best 2 transformations and their weights are reported. The second column shows the number of points chosen by the random selection process. The last column lists a difference in position on the image caused by the difference in the transformation found. Except for the case where no corresponding abstract edges were present, the transformations obtained were very consistent with the "true" transformation and were within the estimated tolerances.

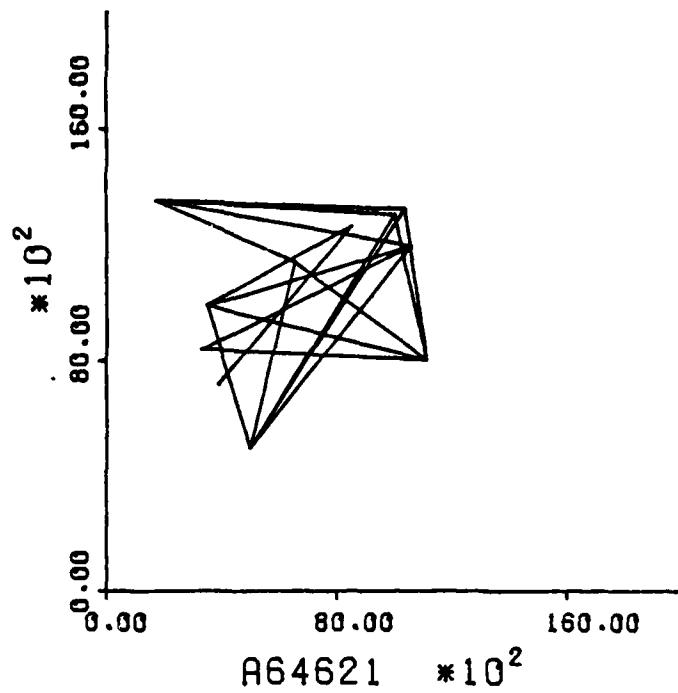
### 3.3.2 Registration of A Stereo Pair

The second set of registration tests used the stereo pair Dreux 12 and Dreux 13, shown in Figures 1.2 and 1.3. As in the case for the image 4621, maps of abstract edges of the two images were created using a Talos digitizer. Registering the known corresponding abstract edges gave the "true" transformation ( $\theta=19\pm1^\circ, \Delta x=-4221\pm30, \Delta y=-1519\pm80$ ).

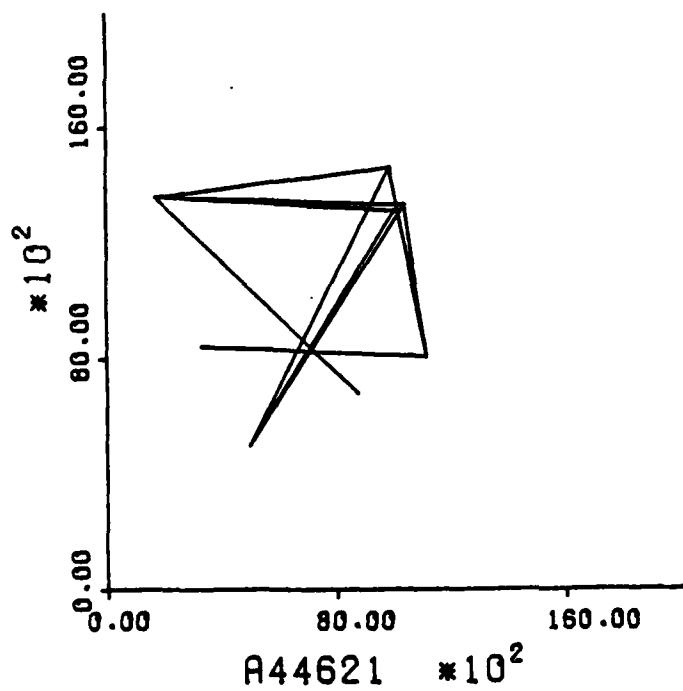
A set of real edges were created for each image using the Talos digitizer and these sets were registered using the registration program. The edges sets are shown in Figure 3.12 and the results of the registration are shown in Figure 3.13. The best transformation was  $(19^\circ, -4207, -1505)$  which is well within the variability of the true transformation. For an image point of (1000, 1000) the difference in transformed positions would be 20 stagels or .010 inches on the image.



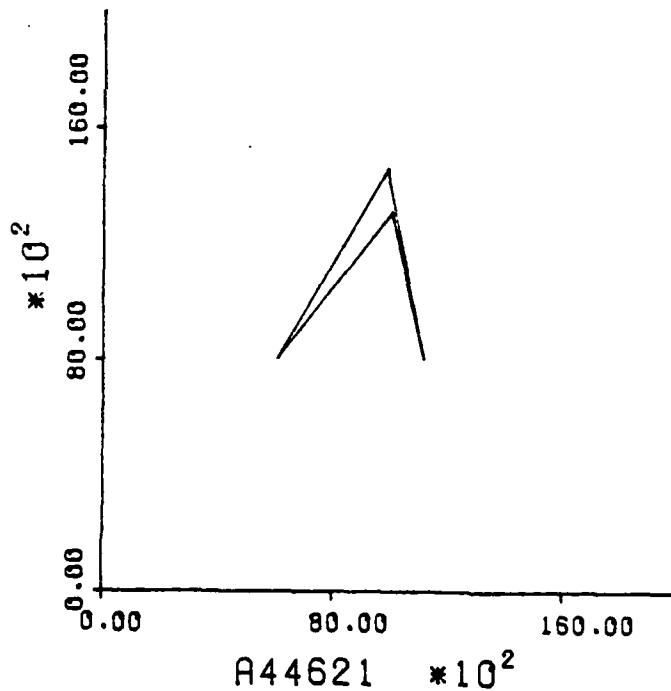
(a) 80% set.



(b) 60% set.

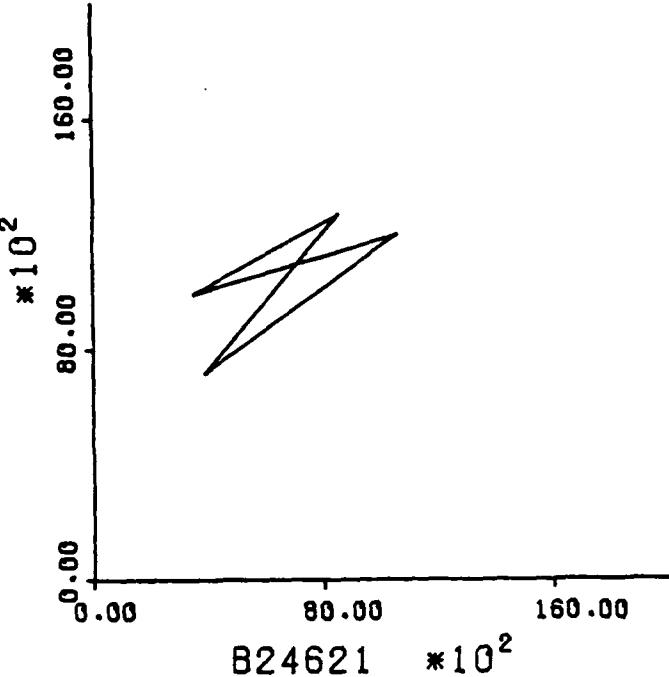


(c) 40% set.

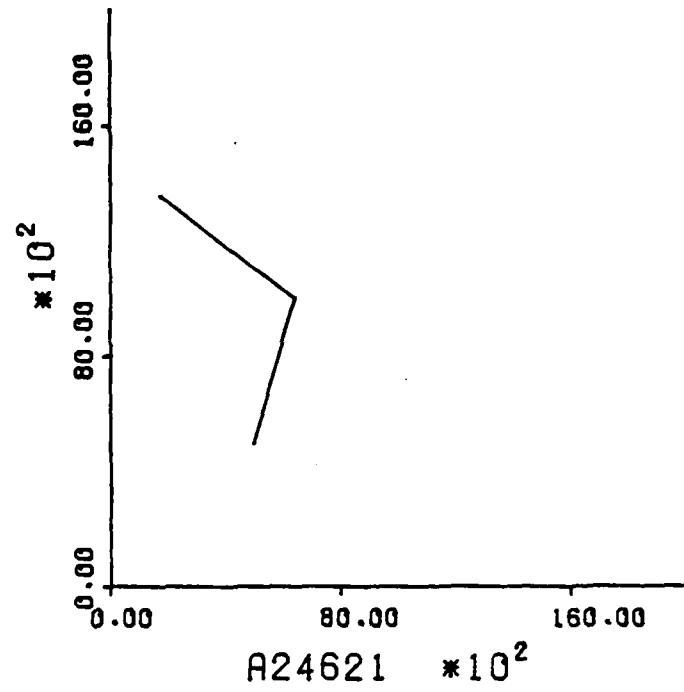


(d) 40% set.

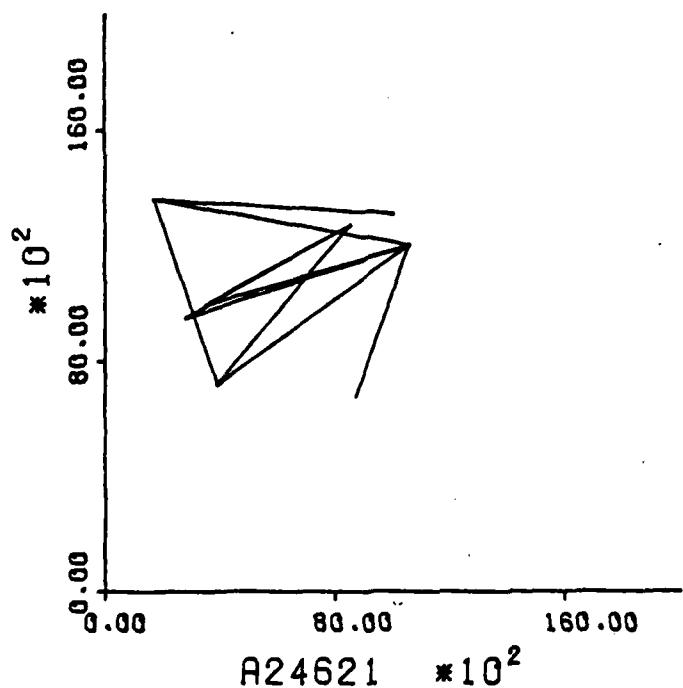
Figure 3.11. Plots of abstract image edges obtained from 4621 under 80-20% reliability criteria.



(e) 20% set.



(f) 20% set.



(g) 20% set.

Figure 3.11 continued

Table 3.2  
Results of Robustness Experiments

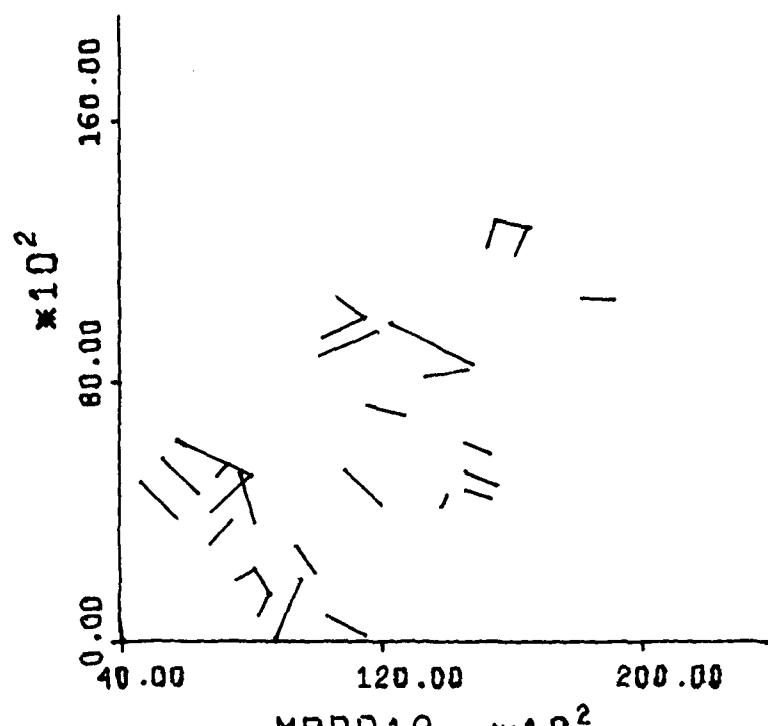
Reliability	Number of * Points Selected	Transformation*** ( $\theta$ , $\Delta x$ , $\Delta y$ )	Weight	Matched Image Edges	Matched Map Edges	Error in Transformation (inches) <sup>†</sup>
80%	27	(309°, 572, -1352) (309°, 598, -1378)	.496 .494	17/23 17/23	18/30 17/30	0.018 0.0
60%	19	(309°, 598, -1378) (309°, 624, -1404)	.494 .492	16/18 16/18	18/30 18/30	0.0 0.019
40%	15	(309°, 572, -1404) (309°, 598, -1378)	.3183 .3176	11/11 11/11	11/30 11/30	0.019 0.0
40%	14	(308°, 598, -1378) (308°, 624, -1352)	.060 .059	2/4 2/4	2/30 2/30	0.0 0.019
20%	8	(309°, 572, -1404) (309°, 520, -1352) (309°, 598, -1378)	.059 .057 .055	2/4 2/4 2/4	2/30 2/30 2/30	0.019 0.041 0.0
20%**	6	(331°, 6032, -2392) (331°, 6084, -2340)	.033 .033	1/2 1/2	1/30 1/30	- ** - **
20%	13	(309°, 598, -1378) (309°, 572, -1404)	.244 .237	7/10 7/10	9/30 9/30	0.0 0.019

\* Total Number of Points was 33.

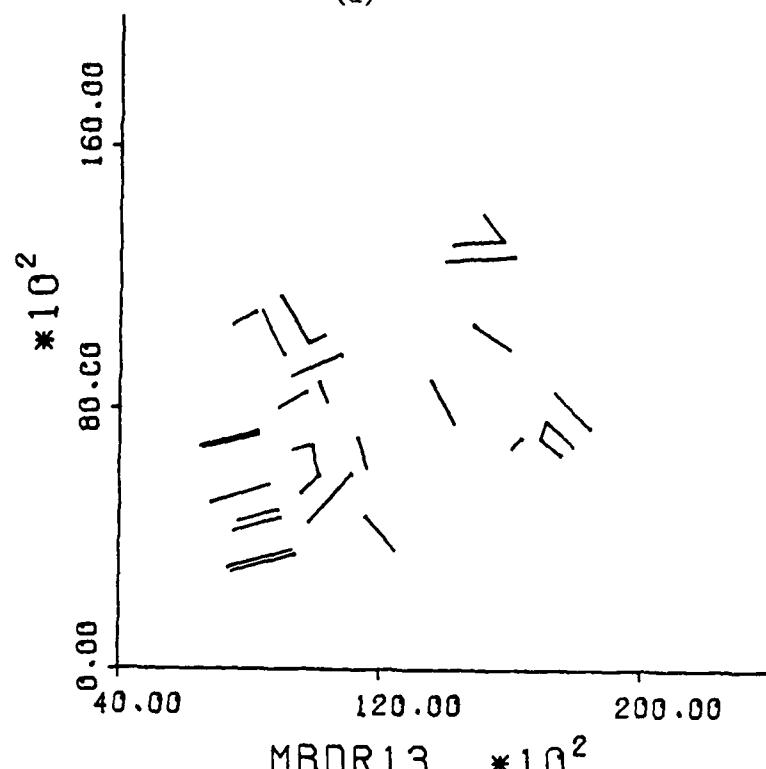
\*\* Neither of the image abstract edges corresponded to any of the map abstract edges.

\*\*\* Variability on transformation parameters were estimated as:  $\theta = \pm 1^\circ$ ,  $\Delta x = \pm 60$ ,  $\Delta y = \pm 40$ .

† For a point at (1000, 1000) on the image.



(a)



(b)

Figure 3.12 Real map edges from (a) Dreux 12 and (b) Dreux 13.

```

INPUT 3,50,200,3,0,0,0,132,0,0,
INPUT 3,50,200,3,0,0,0,132,0,0,
ANGTOL,STOL,DTOL,NUMLEV,PRT:BUK,SBK,MNM, PWIDTH,LENCHK,THRESH =
    3 50 200 3 0 0 0 132 0 0
FOR IMAGE: LOX,LOY,HIX,HIY,XCENTER,YCENTER: 0 01200012000 6000 6000
0 T  SCALE(T)  SCALE(X)  SCALE(Y)
  1  36.000  1200.000  1200.000
  2  3.600   120.000   120.000
  3  .360    12.000    12.000
LOX,LOY,LOTBND(1) = -6000 -6000 0
FOR MAP : LOX,LOY,HIX,HIY,XCENTER,YCENTER: 4000 400018000180001100011000
CLUSTR
CLUSTR
0 NIMAGE,NMAP= 130 30
0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP
  18  1194. -4290. 3513 24/ 130 11/ 30 *** ***
  18  1104. -4308. 3290 24/ 130 11/ 30 *** ***
  18  1116. -4308. 3323 24/ 130 11/ 30 *** ***
0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP
  18  1116. -4188. 3320 25/ 130 11/ 30 *** ***
  18  1116. -4176. 3236 25/ 130 11/ 30 *** ***
  19  1116. -4176. 3526 26/ 130 11/ 30 *** ***
0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP
  20  996. -4284. 2446 20/ 130 10/ 30 *** ***
  20  1008. -4284. 2496 20/ 130 10/ 30 *** ***
  20  1008. -4272. 2566 21/ 130 10/ 30 *** ***
0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP
  274 -4068. -4656. 633 5/ 130 2/ 30 *** ***
  274 -3960. -4668. 546 4/ 130 2/ 30 *** ***
  275 -4068. -4632. 666 5/ 130 2/ 30 *** ***
0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP
  274 -4188. -4704. 593 5/ 130 2/ 30 *** ***
  274 -4176. -4752. 450 4/ 130 2/ 30 *** ***
  274 -4164. -4752. 436 4/ 130 2/ 30 *** ***
0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP
  275 -4188. -4656. 666 5/ 130 2/ 30 *** ***
  275 -4188. -4560. 586 5/ 130 2/ 30 *** ***
  275 -4174. -4656. 666 5/ 130 2/ 30 *** ***
0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP
  19  1212. -4224. 3416 25/ 130 12/ 30 *** ***
  19  1212. -4212. 3413 25/ 130 11/ 30 *** ***
  19  1224. -4212. 3350 25/ 130 11/ 30 *** ***
0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP
  19  1260. -4188. 3083 26/ 130 11/ 30 *** ***
  20  1212. -4140. 2736 21/ 130 9/ 30 *** ***
  20  1212. -4128. 2716 22/ 130 10/ 30 *** ***
0THETA XSHIFT YSHIFT MTCHWT MCHROW/NIMAGE NMCHCL/NMAP
  19  1356. -4164. 2203 21/ 130 9/ 30 *** ***
  19  1368. -4164. 2170 18/ 130 7/ 30 *** ***
  19  1380. -4152. 2190 16/ 130 7/ 30 *** ***
*** END OF CLUSTERING FOR THIS DATA SET ***

```

Figure 3.13 Results obtained from registration software when run with set of real map edges from DREUX 13 and set of real image edges from DREUX 13

### 3.3.3 Conclusions on 2-D Registration

The previous sections presented details on only some of the total number of experiments performed. Although all of these experiments had the same encouraging results, only a limited amount of automatic processing was done and only a small number of images were used. Despite these limitations, the demonstrated robustness of the registration procedure leads us to believe that the automatic feature detection procedures discussed in the later sections of Section 2 are good enough to supply the primitive features needed.

Specific conclusions from the 2-D registration experiments are as follows.

- Registration of stereo images with limited relief can be achieved quite effectively using a linear transformation consisting of only rotation and translation. (Same scale has been assumed.)
- Abstract edges, formed from pairs of accurately determined image points, provide for a fast and accurate registration procedure.
- Use of real edges was successful for registration even when the edge structure barely revealed the image content. However, the registration procedure was much less efficient and accurate than when abstract edges were used.

The registration results are sufficiently good so that no modification appears necessary in the immediate future: effort should be spent on verification, region analysis, and feature detection. However, possible improvements to the registration procedure are as follows.

- The mathematical details relating program tolerances to the amount of local relief in the imagery and the error inherent in the detectors should be worked out to the point of specifying operating procedures. (Tolerances were obtained somewhat arbitrarily in the reported experiments.)

- Thresholds used in clustering should be computed by a strict a priori rule and should be stringently applied so that fewer clusters are examined and less computing done. Similarly, the clustering bin size should be regulated according to the theory of Section 3.2.2.

Very little has been said about the "match matrix" optionally available from the registration program. The match matrix is a matrix whose  $I, J$ -th element is the degree of match between image edge element  $I$  and map edge element  $J$  (i.e.  $EMATCH(I, J)$ ). Since the matrix indicates which edges were in the map but without evidence in the image and vice-versa, it is directly useable for change detection and verification. Verification of features in the map but without edge evidence in the image is reported in Section 5.

### 3.4 Full RST Transformation

The position and orientation of primitive features such as intersections, high curvature points, or lines can accurately be determined, but it is difficult to determine their sizes, such as length. The basic LNK registration technique uses the position and orientation of the primitive features to find the rotation and translation necessary to register two images or to register an image to a map.

If the size could be included, then it would be possible to calculate the four parameter transformation of rotation, translation and scaling. This section presents a method for extending the procedure to account for scaling. In order to accomplish this, abstract vectors or edges whose size can accurately be determined are introduced.

To achieve scaling, abstract vectors or edges can be formed by spanning pairs of point structures. For example, abstract edges can be formed by connecting pairs of high curvature points. Abstract vectors could be formed by using an intersection point as the vector tail and a high curvature point as the vector head. There are many ways of forming the abstract edges or vectors.

The registration procedure is similar to the basic one. Instead of the triples  $(\theta, \Delta x, \Delta y)$ , there are four parameters  $(\theta, \Delta x, \Delta y, \Delta s)$ , where  $\Delta s$  is the scaling parameter.

The three registration steps are now:

- 1) Primitive point features (intersections, high curvature points, etc) are automatically extracted. The abstract vectors are created by pairing the primitive point features. Not all possible pairs need be formed as that would result in a great deal of computation.
- 2) Assume all features of one type can correspond to one another. That is, a vector from an intersection of three lines to an intersection of four lines in the first image can correspond to any 3-intersection to 4-intersection vector extracted from

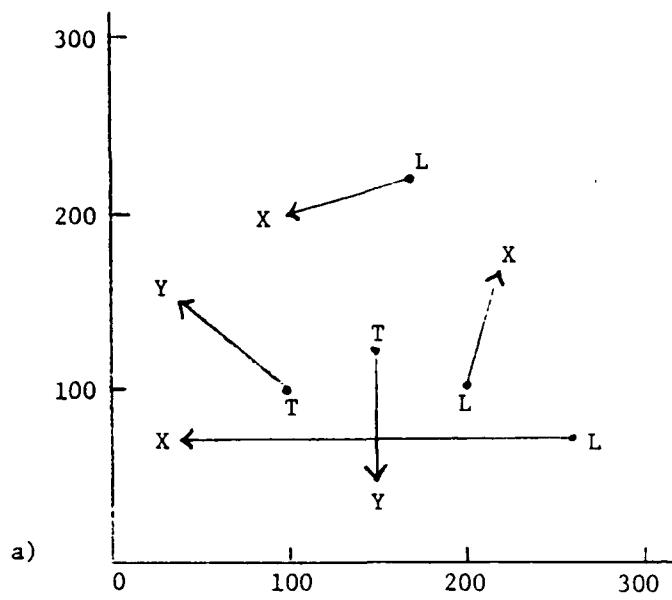
the map (or second image). For each possible correspondence, find the 4-parameter transformation ( $\Delta\theta$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta s$ ) that maps one vector to the other. Place a unit of weight in the bin in the four dimensional histogram that represents the ( $\Delta\theta$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta s$ ) found.

- 3) Locate the best cluster in the histogram. The ( $\Delta\theta$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta s$ ) of that cluster is the best global transformation as it provided the largest number of local correspondences.

As an example of this method, suppose an image and map are represented in terms of vectors  $v_j$  connecting intersection points. The intersection points are of four types; 'L', 'T', 'X', or 'Y'. The rules for pairing points to form vectors may be arbitrary; for example, 'T' points are joined with 'L' points. The purpose of such rules is to control the combinatorics. In Figure 3.14, five vectors represent the map and four vectors represent the image.

Given any map vector  $v_j$  all possible matching image vectors  $v_i$  are considered. Each possible pairing  $(v_j, v_i)$  results in an RS&T transformation mapping image vector  $v_i$  onto map vector  $v_j$ . The transformation is specified by a quad of parameters ( $\Delta\theta$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta s$ ) where  $\Delta\theta$  is the angle of rotation,  $\Delta s$  is the scale factor, and  $\Delta x$  and  $\Delta y$  are the x and y translations respectively. The pair  $(v_j, v_i)$  is discarded without producing a quad if the tips or tails of the vectors  $v_j$  and  $v_i$  disagree in type. In the example of Figure 3.14 there are  $5 \times 4 = 20$  pairs  $(v_j, v_i)$  initially possible and of these only 10 agree in type of tip and tail.

The mathematical development for forming a quad ( $\Delta\theta$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta s$ ) as a function of  $v_j$  and  $v_i$  is given in Figure 3.15. Table 3.3 shows computer output for the example shown in Figure 3.14. There are 10 quads produced and a cluster of size 3 is evident in the neighborhood of the best registration transformation ( $\Delta\theta=5.10$ ,  $\Delta x=-75$ ,  $\Delta y=88$ ,  $\Delta s=0.5$ ). In real world cases there would be hundreds of quads overall and a few dozen in a cluster.



Type Code

L	1
T	2
X	3
Y	4

Map Vectors

$(L, 170, 220) \sim (X, 100, 200)$   
 $(T, 100, 100) \sim (Y, 40, 150)$   
 $(L, 200, 100) \sim (X, 220, 170)$   
 $(L, 260, 70) \sim (X, 40, 70)$   
 $(T, 150, 125) \sim (Y, 150, 50)$

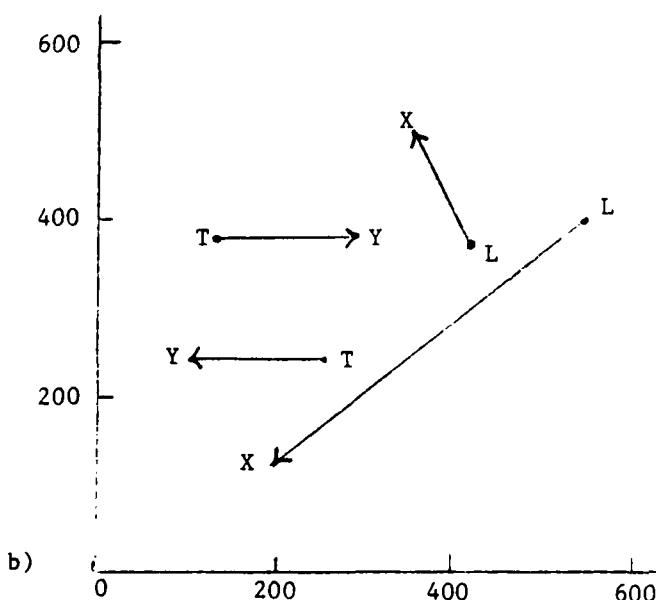
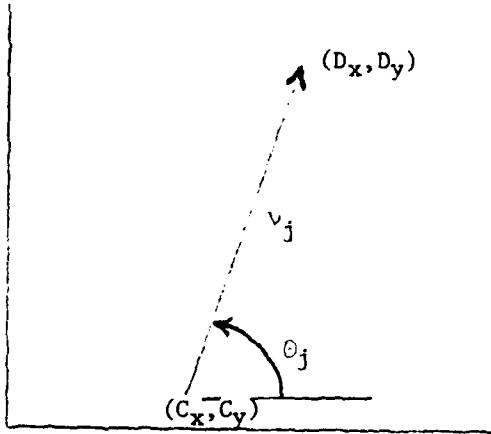
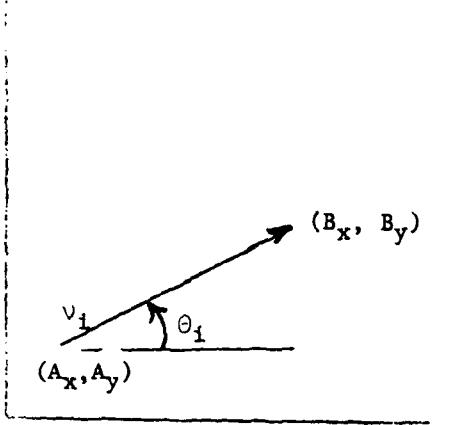


Image Vectors

$(L, 545, 400) \sim (X, 200, 120)$   
 $(T, 260, 240) \sim (Y, 100, 245)$   
 $(T, 140, 380) \sim (Y, 300, 380)$   
 $(L, 420, 370) \sim (X, 360, 500)$

Figure 3.14. a) Example set of map vectors, and

b) set of vectors representing an image to be registered to the map by RS&T transformation.



Assuming that vector  $v_i$  corresponds to vector  $v_j$  transformation parameters  $(\Delta\theta, \Delta x, \Delta y, \Delta s)$  are gotten as

$$\Delta\theta = \theta_j - \theta_i$$

$$\Delta s = \text{length of } v_j / \text{length of } v_i$$

$$\Delta x = \Delta s A_y \sin\Delta\theta - \Delta s A_x \cos\Delta\theta + C_x$$

$$\Delta y = -\Delta s A_x \sin\Delta\theta - \Delta s A_y \cos\Delta\theta + C_y$$

The resulting registration transformation in homogeneous coordinates is

$$[u, v, 1] = [x, y, 1] \begin{bmatrix} \Delta s \cos\Delta\theta & \Delta s \sin\Delta\theta & 0 \\ -\Delta s \sin\Delta\theta & \Delta s \cos\Delta\theta & 0 \\ \Delta x & \Delta y & 1 \end{bmatrix}$$

where  $(x, y)$  is an image point and  $(u, v)$  is the corresponding map point.

Figure 3.15. Mathematical derivation of RS&T transformation parameters from a pair of vectors  $v_i$  and  $v_j$  assumed to be corresponding.

Table 3.3

Ten possible pairings of image and map vectors produce a cluster in RS&T parameter space near ( $\Delta\theta=5.6$ ,  $\Delta x=-75$ ,  $\Delta y=88$ ,  $\Delta s=0.5$ )

MAP VECTOR				IMAGE VECTOR				TRANSFORMATION				
VEC	TAIL	HEAD	VEC	TAIL	HEAD	VEC	TAIL	HEAD	THETA	SCALE	DELX	DELY
#	TYP(X,Y)	TYP(X,Y)	#	TYP(X,Y)	TYP(X,Y)	#	TYP(X,Y)	TYP(X,Y)				
1 1 ( 170, 220) 3 ( 100, 200)	6 1 ( 545, 400) 3 ( 200, 120)								.588+01	.164+00	.421+02	.193+03
1 1 ( 170, 220) 3 ( 100, 200)	9 1 ( 420, 370) 3 ( 360, 500)								.142+01	.508+00	.323+03	-.199+02
2 2 ( 100, 100) 4 ( 40, 150)	7 2 ( 240, 240) 4 ( 100, 245)								.562+01	.488+00	-.721+02	.853+02
2 2 ( 100, 100) 4 ( 40, 150)	8 2 ( 140, 380) 4 ( 300, 380)								.245+01	.488+00	.271+03	.199+03
3 1 ( 200, 100) 3 ( 220, 170)	6 1 ( 545, 400) 3 ( 200, 120)								.375+01	.164+00	.236+03	.205+03
3 1 ( 200, 100) 3 ( 220, 170)	9 1 ( 420, 370) 3 ( 360, 500)								.557+01	.508+00	-.846+02	.967+02
4 1 ( 260, 70) 3 ( 40, 70)	6 1 ( 545, 400) 3 ( 200, 120)								.560+01	.495+00	-.743+02	.863+02
4 1 ( 260, 70) 3 ( 40, 70)	9 1 ( 420, 370) 3 ( 360, 500)								.114+01	.154+01	.506+03	-.754+03
5 2 ( 150, 125) 4 ( 150, 50)	7 2 ( 260, 210) 4 ( 100, 245)								.160+01	.469+00	.266+03	.626+01
5 2 ( 150, 125) 4 ( 150, 50)	8 2 ( 140, 380) 4 ( 300, 380)								.471+01	.469+00	-.281+02	.191+03

### 3.4.1 Example with Scale

The following experiment demonstrates the utility of the proposed technique for regions where cultural activity creates features such as straight edges or networks of lineals. The results reported are typical of many similar experiments.

Point features were identified by eye on an aerial image from the mid-western U.S.A. using two different measuring devices and two different orientations. Figure 1.1 shows sample imagery while Table 3.1 contains the coordinates of the selected feature points. Points are labeled according to the type of road intersection - 'L', 'T', 'X', or 'Y' - which they represent or are labeled 'A' indicating an arbitrary point on the road.

50 vectors were chosen to model the map of Figure 1.1 while 64 vectors represented the image. Tables 3.4 and 3.5 show some of the resulting quads formed by matching vectors from the map with vectors from the image. Note that stage coordinates have been divided by 10 for format convenience. Table 3.4 shows 10 vectors in stage coordinates which have the same type of tip and tail as the vector (4365,2747) - (2498,3641) in the photo model. Only the first of these matches is correct and hence only the first quad (5.39, -1020,4040. 5.01) contributes to the ultimate cluster. Of the  $50 \times 64 = 3200$  pairs  $(v_j, v_i)$  possible only 790 produce quads in cluster space after the check on tip and tail type. Table 3.5 shows that 32 of the 790 quads form a cluster near the parameter set ( $\Delta\theta=5.38$ ,  $\Delta x=-1000.$ ,  $\Delta y=4000.$ ,  $\Delta s=5.0$ ). Using Table 3.5 the reader can verify that 30 of these 32 quads represent correct vector matches. For instance, the first quad represents the matching of vectors from point #2 to point #1 in the two different coordinate systems. The asterisks mark the two incorrect matches which in fact are outliers in the set of 32 quads. A

Table 3.4.

First 10 quads in set of 790 quads produced by comparing 50 map vectors with 64 image vectors.  
(Stage coordinates of Table 4-4 divided by 10 for format convenience).

<u>MAP VECTOR</u>				<u>IMAGE VECTOR</u>				<u>TRANSFORMATION</u>			
VEG	TAIL	HEAD	VEG	TAIL	HEAD	THETA	SCALE	DELX	DELY		
#	TYP(X,Y)	TYP(X,Y)	#	TYP(X,Y)	TYP(X,Y)						
1	3(4365,2747)	4(2498,3641)	51	3( 872, 680)	4( 500, 500)	.539+01	.501+01	-.102+04	.404+04		
1	3(4365,2747)	4(2498,3641)	52	3( 391, 720)	4( 500, 500)	.381+01	.843+01	.322+04	.956+04		
1	3(4365,2747)	4(2498,3641)	53	3( 331, 840)	4( 500, 500)	.380+01	.545+01	.297+04	.747+04		
1	3(4365,2747)	4(2498,3641)	54	3( 281, 951)	4( 500, 500)	.381+01	.413+01	.283+04	.654+04		
1	3(4365,2747)	4(2498,3641)	55	3( 350, 994)	4( 500, 500)	.397+01	.401+01	.237+04	.647+04		
1	3(4365,2747)	4(2498,3641)	56	3( 656,1144)	4( 500, 500)	.450+01	.312+01	.129+04	.549+04		
1	3(4365,2747)	4(2498,3641)	61	3(1001,1310)	4( 500, 500)	.482+01	.217+01	.130+04	.461+04		
1	3(4365,2747)	4(2498,3641)	62	3( 872, 680)	4(1112, 805)	.221+01	.765+01	.125+05	.536+03		
1	3(4365,2747)	4(2498,3641)	67	3( 872, 680)	4( 171,1360)	.324+00	.212+01	.307+04	.793+03		
1	3(4365,2747)	4(2498,3641)	68	3( 391, 720)	4(1112, 805)	.258+01	.285+01	.640+04	.389+04		

Table 3.5

32 quads contained in the bin  $\Theta \in [5.0, 5.7]$ ,  $S \in [4.8, 5.2]$ ,  
 $\Delta x \in [-1100, -900]$ , and  $\Delta y \in [3000, 5000]$ .

<u>MAP VECTOR</u>			<u>IMAGE VECTOR</u>			<u>TRANSFORMATION</u>			
VEC	TAIL	HEAD	VEC	TAIL	HEAD	THETA	SCALE	DELX	DELY
#	TYP(X,Y)	TYP(X,Y)	#	TYP(X,Y)	TYP(X,Y)				
1	3(4365,2747)	4(2498,3641)	51	3( 872, 680)	4( 500, 500)	.539+01	.501+01-.102+04	.404+04	
2	3(3044,4744)	4(2498,3641)	52	3( 391, 720)	4( 500, 500)	.536+01	.501+01-.101+04	.412+04	
3	3(3342,5357)	4(2498,3641)	53	3( 331, 840)	4( 500, 500)	.536+01	.504+01-.103+04	.411+04	
4	3(3611,5908)	4(2498,3641)	54	3( 281, 951)	4( 500, 500)	.537+01	.504+01-.104+04	.408+04	
4	3(3611,5908)	4(2498,3641)	55	3( 350, 994)	4( 500, 500)	.553+01	.489+01-.959+03	.352+04	
6	3(3975,5750)	4(2498,3641)	54	3( 281, 951)	4( 500, 500)	.522+01	.514+01-.994+03	.464+04	
6	3(3975,5750)	4(2498,3641)	55	3( 350, 994)	4( 500, 500)	.538+01	.499+01-.100+04	.406+04	
7	3(5516,5073)	4(2498,3641)	56	3( 656,1144)	4( 500, 500)	.539+01	.504+01-.105+04	.401+04	
8	4(2498,3641)	5(7130,3514)	57	4( 500, 500)	5(1096,1210)	.538+01	.500+01-.101+04	.404+04	
9	1(7430,4091)	4(2498,3641)	58	1(1036,1330)	4( 500, 500)	.538+01	.501+01-.102+04	.407+04	
10	1(7723,4700)	4(2498,3641)	59	1( 986,1460)	4( 500, 500)	.538+01	.495+01-.982+03	.405+04	
11	1(6942,3613)	4(2498,3641)	60	1(1056,1200)	4( 500, 500)	.538+01	.497+01-.992+03	.406+04	
12	3(7233,4181)	4(2498,3641)	61	3(1001,1310)	4( 500, 500)	.538+01	.500+01-.102+04	.406+04	
19	3(4365,2747)	4(5583,2191)	62	3( 872, 680)	4(1112, 805)	.537+01	.495+01-.941+03	.408+04	
20	3(4365,2747)	5(7130,3514)	63	3( 872, 680)	5(1096,1210)	.538+01	.499+01-.994+03	.405+04	
21	1(7430,4091)	3(4365,2747)	64	1(1036,1330)	3( 872, 680)	.537+01	.499+01-.987+03	.410+04	
22	1(7723,4700)	3(4365,2747)	65	1( 986,1460)	3( 872, 680)	.538+01	.493+01-.933+03	.402+04	
24	1(6942,3613)	3(4365,2747)	66	1(1056,1200)	3( 872, 680)	.538+01	.493+01-.923+03	.406+04	
25	3(4365,2747)	4(4861,7575)	67	3( 872, 680)	4( 171,1360)	.538+01	.497+01-.972+03	.406+04	
29	3(3044,4744)	4(5583,2191)	68	3( 391, 720)	4(1112, 805)	.538+01	.496+01-.962+03	.407+04	
30	3(3342,5357)	4(5583,2191)	69	3( 331, 840)	4(1112, 805)	.537+01	.496+01-.956+03	.410+04	
31	3(3611,5908)	4(5583,2191)	70	3( 281, 951)	4(1112, 805)	.537+01	.499+01-.992+03	.410+04	
33	3(3975,5750)	4(5583,2191)	71	3( 350, 994)	4(1112, 805)	.538+01	.497+01-.986+03	.406+04	
35	4(5583,2191)	5(7130,3514)	73	4(1112, 805)	5(1096,1210)	.538+01	.502+01-.105+04	.407+04	
36	1(7430,4091)	4(5583,2191)	74	1(1036,1330)	4(1112, 805)	.537+01	.500+01-.991+03	.414+04	
37	1(7723,4700)	4(5583,2191)	75	1( 986,1460)	4(1112, 805)	.539+01	.494+01-.959+03	.400+04	
38	1(6942,3613)	4(5583,2191)	76	1(1056,1200)	4(1112, 805)	.538+01	.493+01-.927+03	.404+04	
39	3(7233,4181)	4(5583,2191)	77	3(1001,1310)	4(1112, 805)	.537+01	.500+01-.101+04	.410+04	
46	3(3044,4744)	5(7130,3514)	78	3( 391, 720)	5(1096,1210)	.538+01	.497+01-.967+03	.404+04	
47	1(7430,4091)	3(3044,4744)	79	1(1036,1330)	3( 391, 720)	.538+01	.499+01-.991+03	.406+04	
48	1(7723,4700)	3(3044,4744)	80	1( 986,1460)	3( 391, 720)	.538+01	.493+01-.935+03	.406+04	
50	1(6942,3613)	3(3044,4744)	81	1(1056,1200)	3( 391, 720)	.538+01	.495+01-.955+03	.408+04	

4-D clustering routine would need to be implemented to find the significant clusters.

### 3.5 Extending LNK Registration to 3 Dimensions

So far the LNK registration procedures have ignored the 3D aspects of the objects. This section presents a method to extend the 2D registration to 3D under certain constraints. The 2D procedure found RS&T transformations mapping map structure onto image structure. Experiments showed that the RS&T assumption was feasible in cases where variation in the 3rd dimension was relatively small. Good approximate RS&T registration transformations were obtained automatically for photo/map pairs even when there was some relief in the terrain. There are many cases where RS&T transformations are inadequate, such as in low altitude aerial imaging and in acquisition of solid objects by a robot vision system. In these cases projective transformations must be used.

In the general case 6 parameters are necessary to specify imaging in a 3D world [Duda and Hart 1973]. In this section, we consider a constrained imaging system with only 3 free parameters as shown in Figure 3.16. A front image plane is used with reference system origin ( $x=0, y=0, z=0$ ) at the image center. The camera has known focal length  $f$  and looks vertically down at a scene with base distance  $y_0$  from the image plane. There are only 3 unknown parameters to discover; the angle  $\theta$  at which the object lies on the base plane and the amount of translation  $(x_0, z_0)$  of the object origin from the point where the camera axis pierces the plane.

There is some justification for this assumption in the aerial imaging case. First,  $f$  is usually known. Second, it is possible to get a good approximation for altitude  $y_0$  and to achieve a nearly vertical camera axis. These approximations would perhaps be good enough to correctly detect an approximate 3 parameter view which could be used as an initial approximation

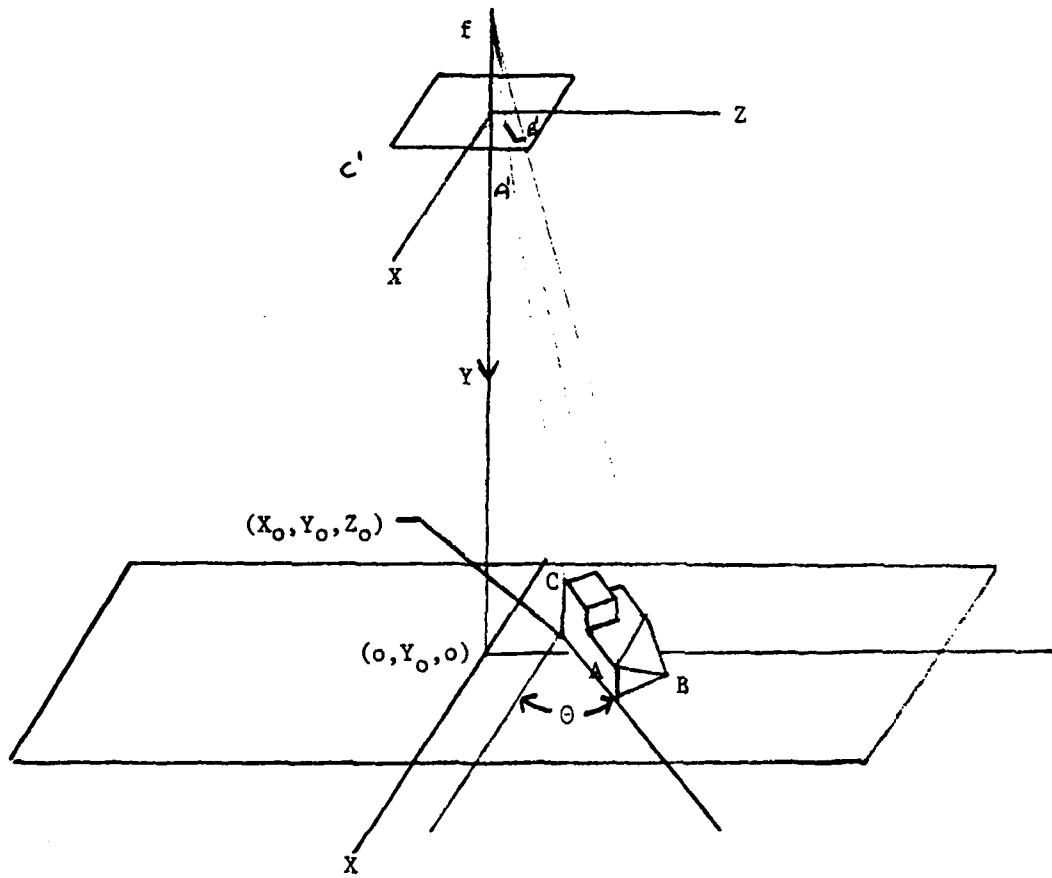


Figure 3.16. Three dimensional object viewed by camera with known attitude. Knowns are  $f$  and  $y_o$  and a model of the object. Unknowns are the object orientation parameters  $\theta$ ,  $x_o$ , and  $z_o$ .

to a full 6 parameter view.

We therefore proceed as follows. A 3D terrain model is specified in local coordinates as in Figure 3.17. The model would contain the location and description of all significant features such as edges, corners, intersections, water bodies, etc.

The acquisition problem is defined as discovering (computing) the orientation parameters  $(\theta, x_o, z_o)$  from the image structure and the known camera parameters  $f$  and  $y_o$ . A few definitions are appropriate before proceeding.

**camera parameters** - parameters that fix the imaging system over the base plane, i.e.  $f$  and  $y_o$ , and define the global coordinate system.

**orientation parameters** - parameters that fix the object (or object model) in the global coordinate system which are  $\theta, x_o, z_o$ .

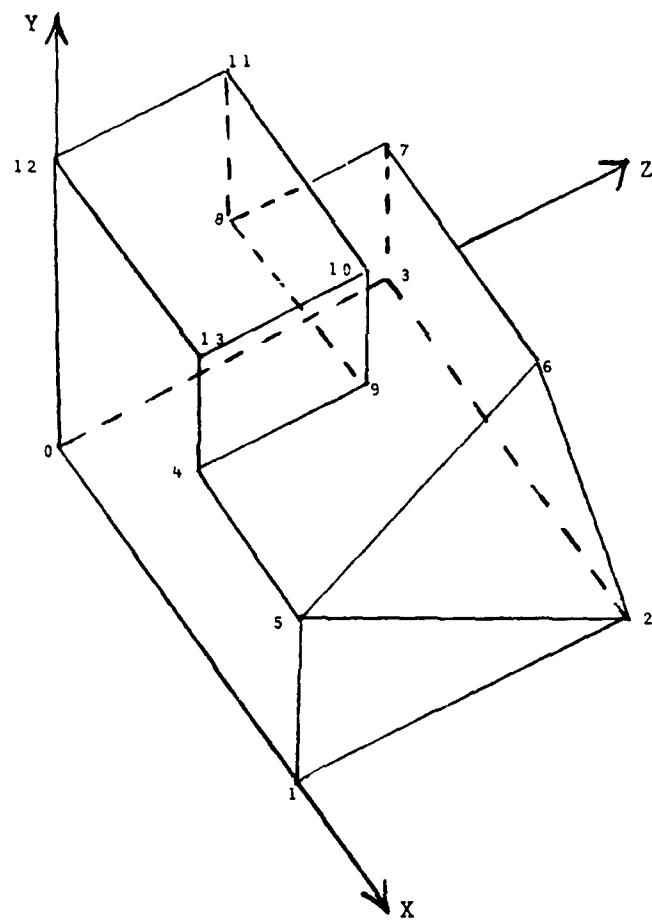
**viewing parameters** - the combined camera and orientation parameters  $f, y_o, \theta, x_o, z_o$  which allow a specific image to be created (from the terrain model).

Here we assume that we know the attitude of the camera  $f, y_o$  and the orientation  $\theta, x_o, z_o$  of the object. We develop computational formulas for image point  $(x_I, y_I, z_I)$  corresponding to point  $(x_m, y_m, z_m)$  in the map.

Let  $(x, y, z)$  be the global coordinates of point  $(x_m, y_m, z_m)$  under map orientation  $(\theta, x_o, z_o)$ . Then we have

$$\begin{aligned} (1) \quad x &= x_m \cos \theta - z_m \sin \theta + x_o && \text{(3D Map to} \\ (2) \quad y &= y_o - y_m && \text{3D Global)} \\ (3) \quad z &= x_m \sin \theta + z_m \cos \theta + z_o \end{aligned}$$

The global coordinates  $(x, y, z)$  then produce image coordinates  $(x_I, y_I, z_I)$



a)

<u>PT</u>	<u>COORDS</u>	<u>PT</u>	<u>COORDS</u>
0	(0,0,0)	7	(0,1,2)
1	(3,0,0)	8	(0,1,1)
2	(3,0,2)	9	(1,1,1)
3	(0,0,2)	10	(1,2,1)
4	(1,1,0)	11	(0,2,1)
5	(3,1,0)	12	(0,2,0)
6	(2,1,2)	13	(1,2,0)

b)

Figure 3.17. Object model defined by a set of vertices and edges.

via the direct perspective transformation.

$$(4) x_I = \frac{fx}{f + y} = \frac{f(x_m \cos \theta - z_m \sin \theta + x_o)}{f + y_o - y_m} \quad (3D \text{ Global})$$

$$(5) y_I = o$$

to  
2D Image)

$$(6) z_I = \frac{fz}{f + y} = \frac{f(x_m \sin \theta + z_m \cos \theta + z_o)}{f + y_o - y_m}$$

Here we assume that a given vector  $(Ax_m, Ay_m, Az_m) - (Bx_m, By_m, Bz_m)$  in the map corresponds to a given vector  $(Ax', Az') - (Bx', Bz')$  in the image. We develop computational formulas for determining map orientation parameters  $(\theta, x_o, y_o)$  from this correspondence.

Rearranging the imaging equations (4) and (6) from above we have the following.

$$(7) \frac{x_I(f - y_m + y_o)}{f} = x_m \cos \theta - z_m \sin \theta + x_o$$

$$(8) \frac{z_I(f - y_m + y_o)}{f} = x_m \sin \theta + z_m \cos \theta + z_o$$

Since (7) must hold for both points A and B, we get two equations from which  $x_o$  can be eliminated, leaving only  $\theta$  unknown.

$$(9) \frac{Ax_I(f - Ay_m + y_o)}{f} - \frac{Bx_I(f - By_m + y_o)}{f} = (Ax_m - Bx_m) \cos \theta + (Bz_m - Az_m) \sin \theta$$

Equation (9) is of the form

$$c = d \cos \theta + e \sin \theta$$

where we make the substitutions  $w = \sin \theta$  and  $\sqrt{1-w^2} = \cos \theta$ .

Thus a standard quadratic equation in  $w$

$$(10) (e^2 + d^2)w^2 - (2ce)w + (c^2 - d^2) = 0$$

is obtained where the coefficients are obtained as follows:

$$c = (Ax_I(f - Ay_m + y_o) - Bx_I(f - By_m + y_o)) /$$

$$d = Ax_m - Bx_m$$

$$e = Bz_m - Az_m.$$

Solving the quadratic yields

$$(11) \quad w = \frac{ce + d\sqrt{e^2 - c^2 + d^2}}{e^2 + d^2}$$

$$(12) \quad \theta = \sin^{-1} w.$$

Knowing  $\theta$ , (7) and (8) can be used to solve for  $x_o$  and  $y_o$  using either A or B point coordinates. For example,

$$x_o = Ax_I(f - Ay_m + y_o) / f - Ax_m \cos \theta + Az_m \sin \theta.$$

Since there is mathematical ambiguity in  $\theta$  from (11) two parameter sets  $(w_1, x_{01}, y_{01})$  and  $(w_2, x_{02}, y_{02})$  can result. It is easy to check for correct parameters using (4) and (6) and the 2 known pairs of corresponding points  $(A_I, A_m)$  and  $(B_I, B_m)$ .

There are two significant cases to note where the computation breaks down. First of all, the discriminant of the quadratic can be negative and hence no  $\theta$  can be obtained. This will happen whenever the map edge cannot possibly be imaged onto the image edge. Few pairings are actually possible due to the fixed scale imposed by  $f$  and  $y_o$ . Secondly, whenever the map edge is vertical both  $d$  and  $e$  above are zero and equations (11) and (12) cannot produce  $\theta$ . Physically we can rotate the object in the map freely about that vertical edge without altering its image and thus we should not expect to get  $\theta$  mathematically either.

Figure 3.18 shows a planar section containing the camera axis, the vertical map vector, and hence the image vector as well. Clearly, free rotation of the object about the axis AB will not change this picture. It is also clear, however, that locational parameters  $x$  and  $z$  are completely specified

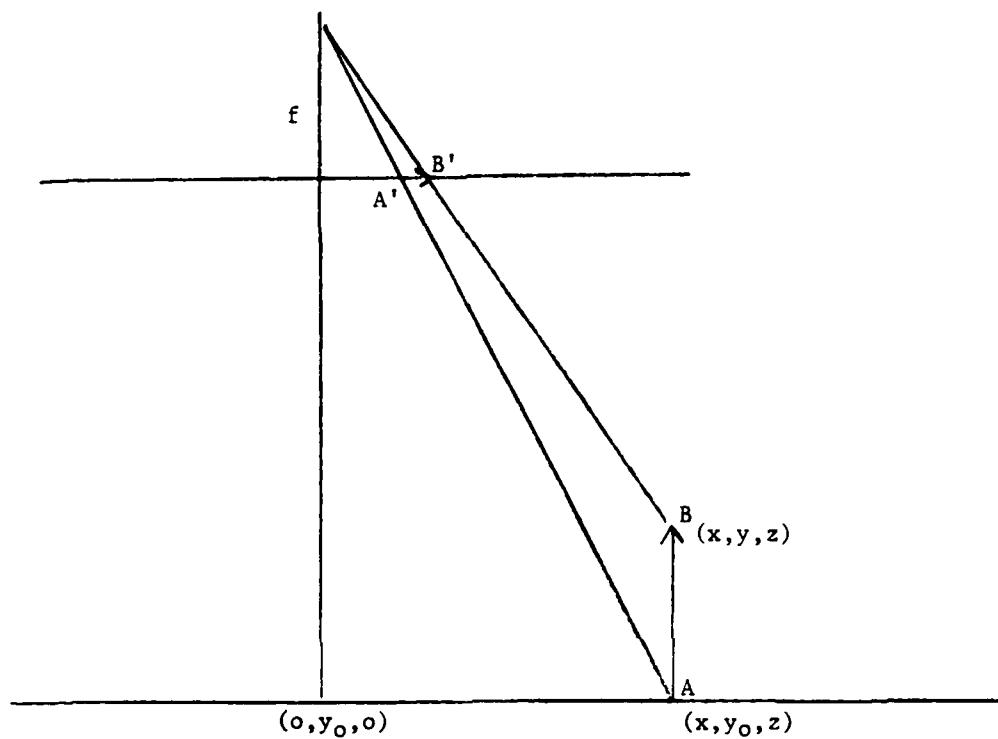


Figure 3.18. Planar section of imaging environment containing both the camera axis and a verticle model vector AB.

by  $A'B'$  and  $AB$  provided that they correctly match via the viewing transformation. From the imaging equations (4) and (6) applied to both points  $A$  and  $B$  we get

$$\begin{aligned} Ax_I(f+y_o) / f &= x = Bx_I(f+y) / f \text{ or} \\ (23) \quad Ax_I(f+y_o) &= Bx_I(f+y) \end{aligned}$$

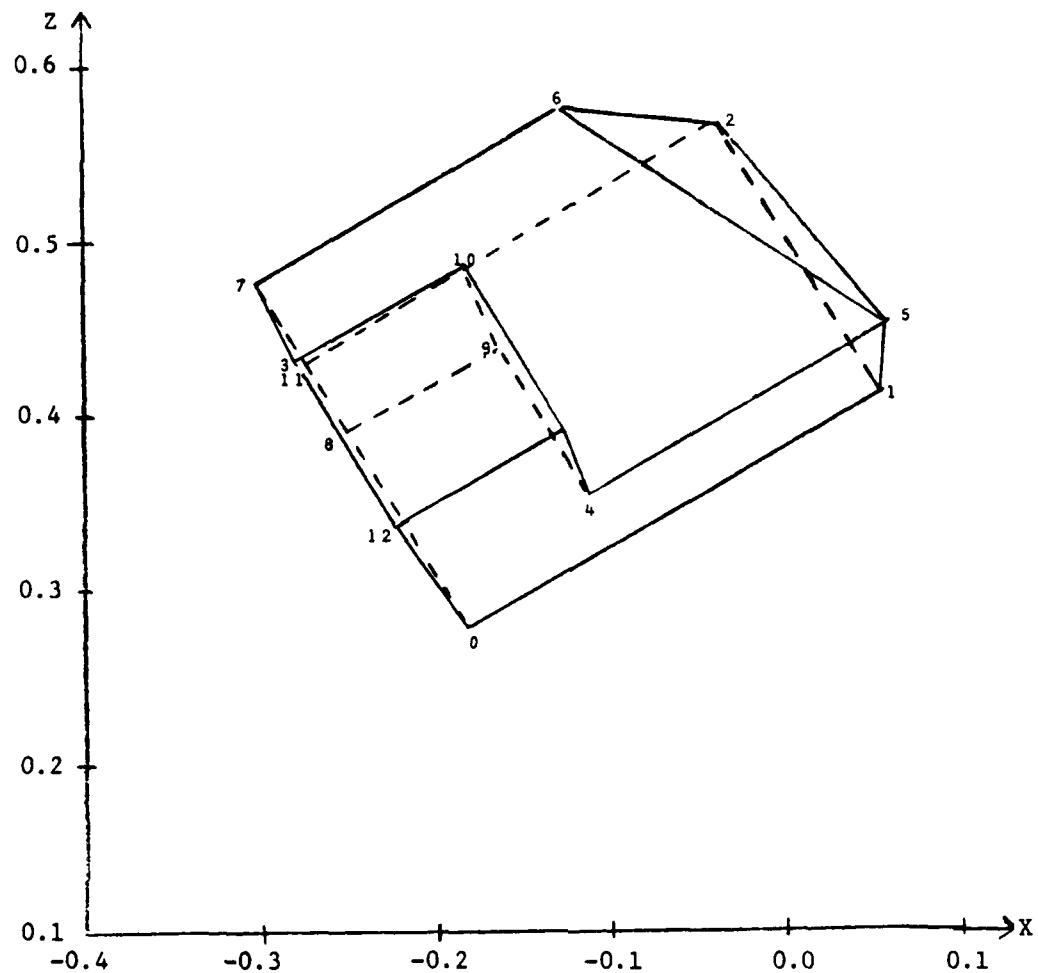
and similarly for the  $z$  coordinate relations

$$(24) \quad Az_I(f+y_o) = Bz_I(f+y).$$

Conditions (23) and (24) must hold if  $A'B'$  is to possibly match with  $AB$ .

We are already in trouble here if only real edge segments are available because (23) and (24) are scaling equations. On the other hand, if abstract vertical map edges which have accurate tips and tails are being used, it would be easy to consider only nonvertical vectors as before. All that need be done is to construct nonvertical vectors by mixing the tip and tail points of several vectors. For this reason, special treatment of vertical vectors can be ignored (justifiably) in the computer programs.

As an example, suppose the map contained the object shown in Figure 3.17. Suppose the image was formed by viewing the object with aspect parameters  $(f=1, y_o=10)$  and orientation parameters  $(\theta=30^\circ, x_o=-2, z_o=3)$ . The resulting image is shown in Figure 3.19, along with the coordinates of the points. Let the object in the map and image be represented by the vectors listed in Table 3.6. Matching the  $10 \times 10$  pairs of vectors  $(v_i, v_j)$  yields only 12 feasible parameter sets, 10 of which form a cluster about the correct parameters  $\alpha = (\theta=0.525, x_o=-2, z_o=3)$ .



<u>Pt</u>	<u>New Coords: (X,Z)</u>
0	(-.182, .273)
1	( .054, .409)
2	(-.037, .567)
3	( .273, .430)
4	(-.113, .350)
5	(-.060, .450)
6	(-.127, .573)
7	(-.300, .473)
8	(-.250, .387)
9	( .163, .437)
10	( .182, .485)
11	(-.278, .430)
12	(-.222, .333)
13	(-.126, .389)

Figure 3.19. Object model defined by set of vertices and edges.

Table 3.6

Representation of model and image by 10 vectors each, chosen from Figures 4-12 and 4-14.

MODEL VECTOR			IMAGE VECTOR			
Number	Tip	Tail	Number	Tip	Tail	
1	(-0.182, .273)	(.054, .409)	PT 0 TO PT 1	1	(0, 0, 0.)	(3, 0, 0.)
2	( 0.054, 0.409)	(-0.037, 0.547)	PT 1 TO PT 2	2	(3, 0, 0.)	(3, 0, 2.)
3	(-0.037, 0.567)	(-0.273, 0.430)	PT 2 TO PT 3	3	(3, 0, 2.)	(0, 0, 2.)
4	(-0.183, 0.273)	(-0.273, 0.430)	PT 0 TO PT 3	4	(0, 0, 0.)	(0, 0, 2.)
5	(-0.113, 0.350)	( 0.060, 0.450)	PT 4 TO PT 5	5	(1, 1, 0.)	(3, 1, 0.)
6	(-0.037, 0.567)	( 0.060, 0.450)	PT 2 TO PT 5	6	(3, 0, 2.)	(3, 1, 0.)
7	( 0.060, 0.450)	(-0.127, 0.573)	PT 5 TO PT 5	7	(3, 1, 0.)	(2, 1, 2.)
8	(-0.127, 0.573)	(-0.300, 0.473)	PT 6 TO PT 7	8	(2, 1, 2.)	(0, 1, 2.)
9	(-0.037, 0.567)	(-0.127, 0.573)	PT 2 TO PT 6	9	(3, 0, 2.)	(2, 1, 2.)
10	(-0.300, 0.473)	(-0.250, 0.387)	PT 7 TO PT 8	10	(0, 1, 2.)	(0, 1, 1.)

#### 4. Region Identification

The interpretation of images frequently necessitates the extraction and identification of significant regions. This requires finding the boundaries of the regions and assigning labels which identify the regions' contents. The segmentation of images into regions was briefly presented in Section 2.8. The identification of regions can be thought of as a partitioning process.

Given a segmentation, the first step of region identification is to find a set of features that would be used to provide an initial set of feasible labels for the regions. This discrimination could be done using statistical or structural pattern recognition procedures. Since a region could be multiply classified, the second part of region identification involves disambiguating the labels using either inter-region relationships and/or structural information from a map.

In Section 4.1, we present statistical methods which use the feature vector to provide an initial set of labels for the regions. Experimental studies comparing several of the methods are described.

In Section 4.2, we present a brief overview of structural methods for reducing the ambiguity of the region labelling. A graph theoretical method of region matching between an image and a map is described. The technique uses network flow analysis to perform the disambiguation. A second method which uses an iterative procedure based on similarity measures between neighboring regions is sketched.

#### 4.1 Region Label Weights

This section provides an overview of the problem of assigning a set of labels and associated likelihoods or weights to these labels given a feature vector describing the region. The labels will be various classes such as forest, field, etc. into which the region can be classified. Ideally the weights should be the conditional probability of the region feature vector given each of the classes. Unfortunately the estimation of these quantities is, in general, very difficult without a large number of samples. The weights described in this section can be obtained by using training samples to form a model for each class and taking as weights some measure of fit of the region feature vector to the model. The models discussed here fall into two categories. First we describe models, called single models, which are defined using only training samples from the class being modelled. Second we describe models, called multi models, which are extracted from classification procedures using training samples from two or more classes simultaneously.

A class model defined solely in terms of the samples from that class has several advantages over a class model based on training samples from several classes. First, the addition of new classes can be easily accommodated by using samples from these new classes to form the new models. The models of the previously existing classes would require no change, thus providing a significant reduction in computation and data storage requirements over multi models. A second advantage of single models over multi models is the improved quality of class fitting. Multi models, since they are based on classification methods, attempt to use variables which are best suited to class discrimination, and use the same variables for all classes being modeled. These multi methods may require the use of variables irrelevant to certain classes thus distorting the

estimation of separating surfaces. Single models, on the other hand, develop class models using only those variables relevant to the description of the class involved. A third advantage of single models over multi models is their behavior in the presence of test samples from new classes for which no training samples are available. In single models the sample from the new class might not fit any model well in which case it could be easily isolated for further examination. Multi methods use the partition of feature space given by the classification procedure and some measure of distance of a sample from each set in the partition. In this case the partition is not adequate to design separation measures which detect the presence of samples from a new class since all samples fall within some set of the partition.

Multi methods can be advantageous in situations where variables good for class separation are present but variables used in the single models contain little information useful for discrimination. Unfortunately, the characterization of data for the purpose of determining the relevance of single versus multi models is not all understood. In Section 4.1.1 to 4.1.5 we examine specific models illustrating by example certain pitfalls with those methods. Section 4.1.1 describes the disjoint principal components model which is a single model. Section 4.1.2 details the pooled covariance discrimination model which is a multi model. In Sections 4.1.3 thru 4.1.5, experiments using these models are described.

#### 4.1.1 Disjoint Principal Components Model

The disjoint principal components model is formed by fitting separate principal components to each class and using the distance from a test sample to the subspace of feature space given by the principal components models as the measure of membership in the class. Thus, smaller weights indicate greater likelihood of class membership. In this section we review the theory of principal components and describe the disjoint principal components methods in detail, indicate some of its limitations and provide suggestions for improvement of the procedure.

The goal of principal components [Kshirsagar 1972] is to give a simple model for a continuous multivariate distribution which maximizes the explained variance of the distribution. This goal has been stated in a very general form since there are at least eight distinct natural optimization questions whose solutions are the principal components. Before describing some of these characterizations, we define principal components and give some basic properties. Let  $x=(x_1 \dots x_p)$  be a sample from a  $p$ -dimensional continuous distribution with mean 0 and covariance matrix  $\Sigma = (\sigma_{ij})$  where  $\Sigma$  is a real positive semidefinite matrix with eigenvalues  $\delta_1, \dots, \delta_p$  where  $\delta_1 \geq \delta_2 \geq \dots \geq \delta_p \geq 0$ . There exists a  $p \times p$  orthogonal matrix  $\Gamma$  whose columns  $\gamma_1, \dots, \gamma_p$  are the eigenvectors of  $\Sigma$  such that  $\Sigma = \Gamma \Delta \Gamma'$  or  $\Gamma' \Sigma \Gamma = \Delta$  and  $\Delta = \text{diag}(\delta_1, \dots, \delta_p)$ , is the matrix with entries  $\delta_1, \dots, \delta_p$  along the main diagonal and zero elsewhere and " $'$ " denotes the transpose of a matrix. For any  $x \in \mathbb{R}^p$ , we define the principal components  $v_1, \dots, v_p$  of  $x$  to be the components of the vector  $v = \Gamma' x$ . Intuitively we have merely performed a change of basis to the basis in which the eigenvectors of  $\Sigma$  are the new basis vectors given in terms of decreasing eigenvalues. Thus  $v_1$  corresponds to the eigenvalue  $\delta_1$ ,  $v_2$  corresponds to  $\delta_2$ , etc. The principal components are uncorrelated and their variances are  $\delta_1, \dots, \delta_p$  respectively.

We now describe several optimal properties of principal components.

First, suppose we wish to find  $q$  ( $q < p$ ) linear combinations of  $x_1, \dots, x_p$  which account for as much of the variation of  $x_1, \dots, x_p$  as possible. Thus we seek a  $q$  by  $p$  matrix  $H$  whose rows  $h_1, \dots, h_q$  are orthonormal such that the sum of the variances of  $h_1'x, \dots, h_q'x$  is maximized. It can be shown that  $h_i = v_i$  for  $i = 1, \dots, q$ . Thus the first  $q$  principal components are the linear combinations of  $x_1, \dots, x_p$  explaining the maximum amount of the variance of  $x_1, \dots, x_p$ . We give one other optimal property of principal components here. Suppose that instead of attempting to explain as much of the variance of  $x$  as possible we attempt to approximate the covariance matrix  $\Sigma$  by a matrix  $B$  of the same order  $p$  and rank  $q$  ( $< p$ ). We wish to find such a  $B$  minimizing

$$\text{Norm}(\Sigma - B) = \left\{ \sum_{i=1}^p \sum_{j=1}^p (\sigma_{ij} - b_{ij})^2 \right\}^{1/2}$$

where  $B = (b_{ij})$ . To state the optimality principle here, we first recall that

$\Sigma$  can be written as

$$\Sigma = \delta_1 \gamma_1 \gamma_1' + \delta_2 \gamma_2 \gamma_2' + \dots + \delta_p \gamma_p \gamma_p'$$

according to the spectral decomposition theorem. It can be shown that the optimal matrix  $B$  is given by

$$B = \delta_1 \gamma_1 \gamma_1' + \dots + \delta_q \gamma_q \gamma_q'$$

This matrix has eigenvectors  $\gamma_1, \dots, \gamma_q$  and eigenvalues  $\delta_1, \dots, \delta_q$  so the principal components computed with this approximate matrix are just the first  $q$  principal components computed from  $\Sigma$ .

Various hypotheses can be tested to determine the number,  $k-1$ , of principal components to select. These tests assume a multivariate normal distribution for the random vector  $x$ . Two hypotheses which can be used are

$$1) \sum_{i=p-k+1}^p \delta_i \text{ is zero.}$$

$$2) \sum_{i=p-k+1}^p \delta_i / \sum_{i=1}^p \delta_i \text{ is zero.}$$

Hypothesis (1) states that the sum of the variances of the last  $k$  principal

components is small while hypothesis (2) states that the ratio of the sum of the last  $k$  variances to the sum of all the variances is zero. Both hypotheses have intuitive appeal as measures of the insignificance of the last  $k$  principal components. Details of these tests are given in [Kshirsagar 1972].

The principal component approach to distribution representation can be adapted to the problems of classification and ranking of class labels [Wold 1976]. We assume  $k$  classes  $C_1, \dots, C_k$  are present. Each class  $C_i$ ,  $i=1, \dots, k$  is represented by a collection of samples where a sample is a point in  $R^n$ . First all variables in each class are modified to have zero mean and unit variance. This is done to eliminate the effect of magnitude difference between variables due to inappropriate choices of scales for the various variables. For each class,  $C_i$ , a principal component model is formed using the normalized variables. The number of principal components may be manually or automatically selected using any of a range of tests [Kshirsagar 1972]. Having formed these models a sample can be tested for its fit to each of the models. Let  $m_{ij}$  and  $s_{ij}$  denote the unnormalized mean and variance of the  $j$ th variable of the  $i$ th class. Let  $x = (x_1, \dots, x_n)$  be a sample to be fitted. Define a class  $i$  normalized vector  $x' = (x'_1, \dots, x'_n)$  of  $x$  by  $x'_j = \frac{x_j - m_{ij}}{s_{ij}}$ . Let  $x'' = (x''_1, \dots, x''_n)$  denote the projection of  $x'$  on the space spanned by the eigenvectors corresponding to the selected principal components for this class. Finally we define the fit of  $x$  to class  $i$  to be the Euclidean distance between the vector  $x$  and  $x''$ . Thus we have a measure of the distance of a sample from each class. In this model a class is represented by a linear subspace of  $R^n$ . The advantages attributed to single models in section 4.1 apply to the disjoint principal components model. The selection of a linear subspace to represent a class can be lead to problems in classifying a sample if the sample lies close to a representing subspace but far from the samples used to select this subspace. Before suggesting a modification of the algorithm to overcome these

difficulties, if they arise, we remark that the ranking of classes for a sample is defined by the distance from a class in the obvious way: class  $C_i$  is more likely than class  $C_j$  if the normalized sample is closer to the model for  $C_i$  than it is to the model for  $C_j$ .

A refinement of the principal components models may be obtained by modification of the notion of the distance from a sample to a class. In the subspace chosen by the disjoint principal components method to model a class we may perform a parametric or non-parametric density estimation for the projections of the samples of the class onto the subspace. If the number of principal components is small, this estimate may be quite reliable. The amended notion of the distance from a sample to a class should be small if the sample is near a point in the subspace which has a high probability according to the density estimate. If we normalize the distance of a sample to each class by dividing the ordinary principal components distance to a class by the sum of the distances to all the classes, and we denote this normalized distance by  $d$ , then we can define the new distance of a class by  $d + \frac{1}{p}$  where  $p$  is the estimated probability of the projection of the sample. If we wish to use inverse distances, so that smaller values represent better fits, we may use  $\frac{1}{d} + p$ . In the case of equal prior probabilities for each class and all principal components being used this is just Bayes rule.

#### 4.1.2 Pooled Covariance Discrimination

The posterior probabilities of a sample belonging to each of several classes can be used to rank the likelihoods of membership in each of the classes. If each class is assumed to have a multivariate normal distribution and all classes have a common covariance matrix, then the estimation of the posterior probabilities is considerably simplified. Software for the computation of these probabilities is available in the BMDP package [Dixon 1979]. We now describe the computation of these quantities. For further details, see [Dixon 1979]. Let  $p$  denote the number of variables,  $g$  the number of classes used for design,  $t$  the total number of classes and  $x_{ijr}$  the value of variable  $r$  of sample  $j$  of class  $i$ . Furthermore let  $n$  be the total number of samples and let  $n_i$  be the number of samples in class  $i$ . In the experimental work, each class was equally divided into two classes, one for designing and one for testing. Thus  $t=2g$ . First compute the group means

$$\bar{x}_{ir} = \frac{\sum_{j=1}^{n_i} x_{ijr}}{n_i} \quad i=1, \dots, t \\ r=1, \dots, p$$

and the pooled within group sums of cross-product deviations

$$w_{rs} = \sum_{i=1}^g \sum_{j=1}^{n_i} (x_{ijr} - \bar{x}_{ir})(x_{ijs} - \bar{x}_{is}) \quad r=1, \dots, p \\ s=1, \dots, p$$

A  $qxq$  submatrix,  $W_{11}$ , of the matrix,  $W$ , whose  $ij$ th entry is  $w_{ij}$  is defined by a stepwise procedure. The stepwise inclusion of variables is guided by the  $F$  approximation to Wilk's  $\Lambda$  statistic. Let  $A=W_{11}^{-1}$  and let  $(A)_{ij}=a_{ij}$ . The squared Mahalanobis distance of sample  $j$  in class  $i$  from the mean of group  $k$  is given by

$$D_{ijk}^2 = (n-t) \sum_{r=1}^q \sum_{s=1}^q (x_{ijr} - \bar{x}_{kr}) a_{rs} (x_{ijs} - \bar{x}_{ks}) \\ i=1, \dots, t \\ j=1, \dots, n_i \\ k=1, \dots, g$$

Finally the posterior probability that sample  $j$  from class  $i$  comes from class  $k$  is given by

$$p_{ijk} = p_k \exp(-1/2 D_{ijk}^2) / \sum_{r=1}^g p_r \exp(-1/2 D_{ijr}^2)$$
$$i=1, \dots, t$$
$$j=1, \dots, n_i$$
$$k=1, \dots, g$$

The effectiveness of this approach is dependent on the assumptions of normality and equivalence of the covariance matrices.

#### 4.1.2 Pooled Covariance Discrimination

The posterior probabilities of a sample belonging to each of several classes can be used to rank the likelihoods of membership in each of the classes. If each class is assumed to have a multivariate normal distribution and all classes have a common covariance matrix, then the estimation of the posterior probabilities is considerably simplified. Software for the computation of these probabilities is available in the BMDP package [Dixon 1979]. We now describe the computation of these quantities. For further details, see [Dixon 1979]. Let  $p$  denote the number of variables,  $g$  the number of classes used for design,  $t$  the total number of classes and  $x_{ijr}$  the value of variable  $r$  of sample  $j$  of class  $i$ . Furthermore let  $n$  be the total number of samples and let  $n_i$  be the number of samples in class  $i$ . In the experimental work, each class was equally divided into two classes, one for designing and one for testing. Thus  $t=2g$ . First compute the group means

$$\bar{x}_{ir} = \frac{\sum_{j=1}^{n_i} x_{ijr}}{n_i} \quad i=1, \dots, t \\ r=1, \dots, p$$

and the pooled within group sums of cross-product deviations

$$w_{rs} = \sum_{i=1}^g \sum_{j=1}^{n_i} (x_{ijr} - \bar{x}_{ir})(x_{ijs} - \bar{x}_{is}) \quad r=1, \dots, p \\ s=1, \dots, p$$

A  $qxq$  submatrix,  $W_{11}$ , of the matrix,  $W$ , whose  $ij$ th entry is  $w_{ij}$  is defined by a stepwise procedure. The stepwise inclusion of variables is guided by the  $F$  approximation to Wilk's  $\Lambda$  statistic. Let  $A=W_{11}^{-1}$  and let  $(A)_{ij}=a_{ij}$ . The squared Mahalanobis distance of sample  $j$  in class  $i$  from the mean of group  $k$  is given by

$$D_{ijk}^2 = (n-t) \sum_{r=1}^q \sum_{s=1}^q (x_{ijr} - \bar{x}_{kr}) a_{rs} (x_{ijs} - \bar{x}_{ks}) \\ i=1, \dots, t \\ j=1, \dots, n_i \\ k=1, \dots, g$$

#### 4.1.3 Synthetic Data Experiments

Simulation studies were done to compare the performances of class labelling using pooled covariance density estimation and disjoint principal components. The purpose of these studies was to determine which of these algorithms is more suitable for providing a ranking of class labels. While these experiments shed some light on the characteristics of the methods involved, no firm guidelines can be given for selecting a method except in extreme cases.

Fifteen simulations were performed. The data set for each simulation consisted of four classes, of one-hundred 3-dimensional vectors each. The means of the four classes are  $(0,0,0)$ ,  $(1,0,0)$ ,  $(0,1,0)$ , and  $(0,0,1)$ . The fifteen simulations can be divided into five groups of three simulations each. Within each group of three simulations the same group of four covariance matrices was used for each of the three trials, but different sets of random vectors were generated from these distributions. None of the covariance matrices for these fifteen trials is a diagonal matrix.

For each simulation, fifty of the one-hundred samples from each class were used for designing decision procedures and fifty were used for testing. For each of the 200 test samples, twelve numbers were calculated. First the true density function of each class was used to generate the probability of the sample coming from each class. The probability of each sample belonging to each of the classes was then estimated using the pooled sample covariance matrix over all classes. Finally the distances of the same from the principal component model for each class was generated. The principal component distances were converted into weights in the following way. Let  $d_1, d_2, d_3, d_4$  be the four distances for a sample. To each  $d_i$ ,  $i=1, \dots, 4$ , we correspond the weight  $w_i = \frac{1/d_i}{1/d_1 + 1/d_2 + 1/d_3 + 1/d_4}$ ,  $i=1, \dots, 4$ . Before computing these weights, any  $d_i$  such that  $d_i < .0001$  was replaced by .001. Using this normalization procedure, all weights lie between zero and one and larger weights

correspond to better class fits. For each sample, the three sets of four numbers were converted to rankings. Thus for each sample and each of the three methods of assigning weights we define a sequence  $a_1, a_2, a_3, a_4$  of four distinct integers chosen from the set  $\{1, 2, 3, 4\}$  as follows. Let  $p_1, p_2, p_3, p_4$  be the weights or probabilities of the sample belonging to classes 1, 2, 3, 4 respectively. Then  $a_i=j$  if  $p_j$  is the  $i$ th largest of the numbers  $p_1, p_2, p_3, p_4$ . For each simulation the confusion matrix for each method was computed. These matrices are shown in Figure 4.1. The covariance matrices are given in Figure 4.2. Since the class ordering given by the true probabilities is, on the average, optimal, this ordering was compared with the ordering given by the estimated probabilities and disjoint principal components. In Figure 4.3, the number of times each of these two orderings agreed with the ordering determined by the true probabilities for the top choice, the top two choices, and all four choices is given. Finally in Figure 4.4, the number of times these two orderings have the first two classes correct or reversed is given. The next section discusses the results.

Simulation No	True probability				Estimated Probability				Principal Components			
1	25	12	9	4	19	11	10	10	28	6	9	7
	5	44	0	1	3	47	0	0	6	42	0	2
	6	0	44	0	6	0	43	1	14	1	29	6
	5	0	10	35	4	1	11	34	7	0	6	37
2	37	6	4	3	33	6	7	4	35	5	8	2
	3	44	0	3	6	34	0	5	16	32	0	2
	2	0	48	0	2	1	47	0	3	0	46	1
	5	3	5	37	2	3	8	37	14	1	3	32
3	28	9	6	7	25	7	9	9	35	7	5	3
	4	42	0	4	5	40	0	5	14	32	0	4
	0	0	45	5	6	2	38	4	4	2	39	5
	5	0	2	43	3	0	3	44	5	0	4	41
4	33	5	6	6	25	2	11	12	27	2	12	9
	2	43	0	5	0	43	0	7	6	37	0	7
	5	0	40	5	3	0	45	2	6	0	43	1
	6	2	2	40	6	1	5	38	10	0	2	38
5	32	6	6	6	28	8	1	13	30	6	4	10
	5	40	0	5	2	40	0	8	5	36	0	9
	8	0	38	4	8	0	38	4	7	0	40	3
	1	5	7	37	1	4	10	35	4	2	7	37
6	28	3	7	12	29	2	8	11	26	5	8	11
	3	45	0	2	4	42	0	4	3	42	0	5
	5	0	41	4	7	0	42	1	6	2	40	2
	7	4	2	37	7	6	7	30	11	1	4	34
7	24	11	9	6	22	7	10	11	9	16	11	14
	4	29	9	9	10	25	6	9	9	29	8	4
	5	9	36	0	11	5	34	0	3	11	31	5
	4	8	5	33	6	9	4	31	3	10	5	32
8	12	15	13	10	15	13	11	11	13	15	14	8
	6	36	3	5	7	30	4	9	4	34	2	10
	4	8	35	3	7	7	34	2	9	8	27	6
	9	5	9	27	8	4	9	29	13	9	5	23
9	14	21	6	9	15	18	6	11	13	22	5	10
	6	31	4	9	11	31	0	8	9	27	5	9
	5	4	39	2	10	4	34	2	8	5	36	1
	7	4	5	34	4	4	5	37	9	7	4	30

Figure 4.1 Confusion Matrices for fifteen simulations using true probabilities, estimated probability and disjoint principal components for classification. Rows correspond to true classes and columns correspond to classification results.

Simulation No	True Probability				Estimated Probability				Principal		Components	
10	18	16	8	8	15	13	11	11	14	15	10	11
	7	33	9	1	7	30	11	2	10	30	10	0
	2	2	34	7	6	2	36	6	5	3	34	8
	2	1	4	43	3	2	2	43	3	5	4	38
11	14	7	11	18	11	8	10	21	15	11	9	15
	3	34	10	3	5	35	8	2	6	35	7	2
	3	5	32	10	4	6	30	10	5	6	32	7
	3	1	4	42	4	2	3	41	9	2	6	33
12	11	15	8	16	14	12	9	15	11	11	13	15
	5	38	6	1	4	40	5	1	6	36	8	0
	6	2	33	9	7	4	29	10	6	3	31	10
	7	2	6	35	13	3	2	32	7	5	8	30
13	24	14	6	6	21	12	8	9	12	13	14	11
	8	36	4	2	9	36	1	4	6	34	5	5
	11	6	25	8	6	8	24	12	8	5	24	13
	4	9	3	34	3	7	3	37	4	10	4	32
14	24	16	8	2	14	16	10	10	7	15	13	15
	8	33	3	2	6	30	4	10	4	31	4	11
	10	4	27	6	7	3	30	10	6	3	32	9
	10	8	5	9	6	7	8	29	8	9	5	28
15	18	10	10	12	15	13	12	10	16	10	13	11
	9	29	4	8	8	30	4	8	8	26	8	8
	10	7	31	2	8	9	30	3	9	8	27	6
	12	5	7	26	4	8	9	29	9	7	7	27

Figure 4.1 (cont)

Denote the covariance matrix by  $A = \begin{pmatrix} a_1 & a_2 & a_4 \\ a_2 & a_3 & a_5 \\ a_4 & a_5 & a_6 \end{pmatrix}$

		$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
SET 1	Class 1	.8	.7	.9	.7	.6	.9
	2	.9	.8	.9	.3	.5	.8
	3	.9	.8	.9	.3	.5	.8
	4	.9	.4	.7	.6	.1	.7
SET 2	1	.79	.7	.85	.7	.6	.9
	2	.9	.8	.9	.3	.5	.8
	3	.79	.7	.85	.7	.6	.9
	4	.9	.4	.7	.6	.1	.7
SET 3	1	1.1	.1	.6	.2	.3	.7
	2	.8	.3	.7	.2	.4	.7
	3	.8	.1	.9	.2	.7	.9
	4	.9	.1	.9	.1	.1	.9
SET 4	1	.7	.2	.9	.3	.2	.8
	2	.9	.1	.7	.6	.3	.8
	3	.7	.3	.9	.3	.2	.7
	4	.9	.1	.7	.6	.3	.8
SET 5	1	.9	.2	.9	.1	.4	.7
	2	.9	.3	.7	.3	.3	.9
	3	1.2	.3	1.1	.3	.2	.95
	4	1.2	.3	1.1	.3	.2	.95

Figure 4.2 Covariance matrices for classes used in classification experiments. Each set consists of a set of four covariance matrices corresponding to four classes.

<u>Simulation No.</u>	<u>Estimated Probabilities</u> (# of classes ranked correctly)			<u>Disjoint Principal Components</u> (# of cases ranked correctly)		
	1	2	4	1	2	4
1	172	105	87	155	87	35
2	176	115	84	158	93	24
3	171	109	91	166	101	37
4	173	102	79	163	98	42
5	168	102	95	167	94	53
6	177	110	94	164	97	41
7	167	119	92	142	73	56
8	175	123	81	152	71	41
9	168	120	94	149	93	54
10	174	137	110	168	143	84
11	182	161	130	166	121	73
12	171	128	103	164	124	72
13	163	98	88	157	99	69
14	170	113	87	151	106	82
15	174	121	86	146	88	62
MEAN	172.1	117.5	93.4	157.9	99.2	55
STANDARD DEVIATION	4.6	16.0	12.9	8.4	18.7	18.3

Note: Any sample with at least the first class correct is counted in column one. Any sample with at least two classes correct is counted in column 2.

Figure 4.3 Number of correct rankings (e.g. most likely class is correct, two most likely classes are correctly predicted by the given method) for each of the fifteen simulation experiments.

<u>Simulation No.</u>	<u>Estimated Probabilities</u> (Count)	<u>Disjoint Principal Component</u> (Count)
1	119	108
2	130	118
3	126	123
4	123	118
5	123	112
6	127	118
7	143	100
8	142	88
9	143	129
10	149	154
11	175	136
12	146	145
13	123	119
14	133	136
15	140	124
Mean	136.1	121.9
Standard Deviation	14.5	16.9

Figure 4.4 Number of times first two classes are correct or reversed for each of the fifteen simulation experiments.

#### 4.1.4 Discussion of Region Weight Evaluation

The simulation experiments indicated no advantage of disjoint principal components over density estimation. No theory exists for predicting the relative performance of these methods. A greater variety of covariance matrices should be examined to determine conditions under which each method dominates. Once an adequate supply of examples in which each method dominates is collected, an attempt should be made to relate properties of the distributions to the success of the various methods. One possible distribution property is the similarity of the individual class covariance matrices to the pooled covariances matrix. A crude but simple measure of this distance may be obtained by viewing an  $n \times n$  covariance matrix as a point in  $R^{n^2}$ . The similarity measure can be taken to be the sum of the distances from the individual class covariance matrices to the pooled covariance matrix. A second measure of covariance matrix similarity may be obtained by comparing the estimated posterior probability of each sample based on its class covariance matrix and the estimated probability based on the pooled covariance matrix. The sums of the differences of these quantities may be used as a measure of similarity. Neither of the above measures directly takes into account directional differences in the structure of the distributions. A simple measure of this might be the expected angle (over all classes) between the first principal directions for each pair of classes, where by principal direction we mean the eigenvector corresponding to the first principal component. Numerous other simple measures could be defined. Correlations between these measures and the success of one method over another could then serve as guidelines in the selection of an appropriate data dependent procedure for real data.

#### 4.1.5 Discussion of Region Label Rankings

The comparison of methods for ranking region labels is difficult to perform on image data since the only standard for evaluation is where the true class is located in the rankings. Definitions of class similarity could be defined in terms of the confusion matrix of a classification procedure such as a decision tree. Unfortunately, the evaluation of rankings in terms of these similarity definitions is muddled by the somewhat arbitrary choice of a standard for ranking. The disjoint principal components (DPC) procedure was performed on a test set of one-hundred and twenty samples from the aerial image, frame 4621, supplied by USAETL. These samples were divided into two groups, sixty forest samples and sixty field samples. Each sample consisted of the thirty-two rings representing ROSA spatial frequency data.

In all DPC tests, thirty samples from each class were used for the design set and thirty samples for the test set. Due to the small number of samples used, a subset of the thirty-two rings were selected for application of the DPC procedure. For this subset of selected features, the resulting classification are given in Fig. 4.5. Rings 1, 2, 3, 30, 31, 32 were useful as a starting set of features for DPC. In addition, the density estimation procedure described in Section 4.1.2 was applied to this data. For this work, samples from each class were divided into two groups of approximately equal size using a random procedure to places samples in design or test sets. Each sample had a .5 probability of being placed in the design or the test set. The results of this computation are given in Figure 4.6. The histogram of the first principal component of the samples, using the pooled covariance matrix, is given in Figure 4.7. From this Figure we see that these classes separate very well using the first principal component to represent the data.

Forest versus Field Discrimination

Confusion Matrix  
Density Estimation Using a Pooled Covariance Matrix

Group	Number of Cases Classified Into		Percent Correct
	Field	Forest	
Field	31	2	93.3
Forest	0	25	100

Figure 4.5 Forest versus Field Discrimination (Density Estimation).

Forest versus Field Discrimination

Confusion Matrix  
Disjoint Principal Components (1 component)

Group	Number of Cases Classified Into		Percent Correct
	Field	Forest	
Field	29	1	
Forest	3	27	

Figure 4.6 Forest versus Field Discrimination (Disjoint Principal Components).

Legend:	Design Set		Test Set
	Field	Forest	
		A	C
		B	D

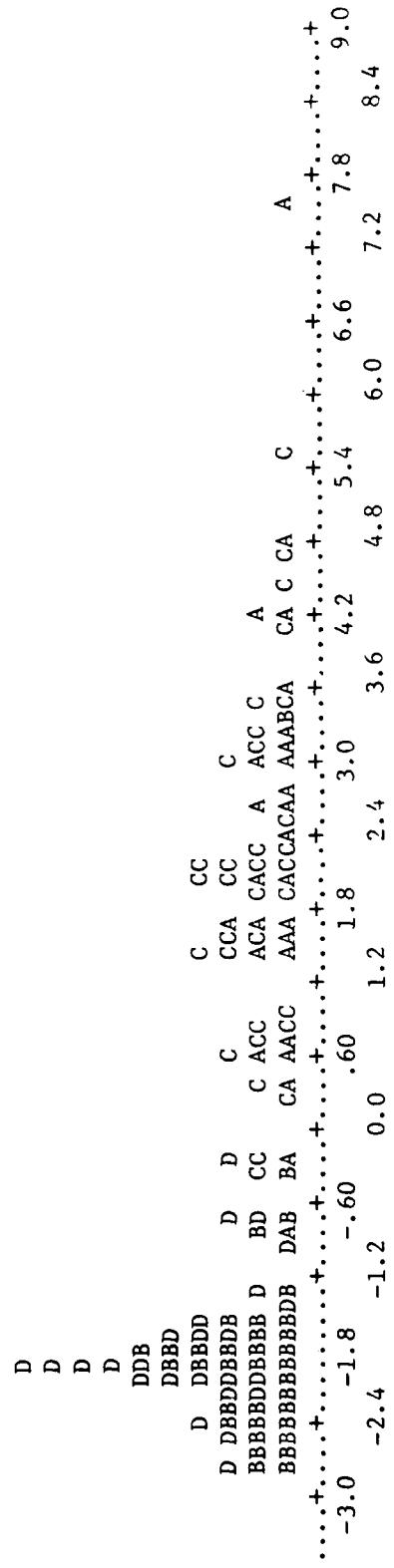


Figure 4.7 Histogram of the First Principal Component of the Pooled Covariance Matrix for the Field Forest Data

#### 4.2 Overview of Region Matching

The classification of regions in an image is a difficult task which sometimes requires application of several levels of preprocessing. In this section we describe two procedures for taking a tentative set of region labels and using region features to obtain a less ambiguous labelling. In addition we describe some experiments indicating a framework for a detailed study of one of these methods using real image data.

#### 4.2.1 Region Image Matching Using Similarity of Region Features

Region image matching can be done by finding a set of features which describe the regions and then pairing regions which have the best matching set of features. It is desirable that the decisions take into account the adjacency information of the regions, so that adjacent regions in the image match adjacent regions in the map.

One such method of region matching yields a measure of similarity between pairs of regions, one from each of the two images to be matched [Price and Reddy 1979]. This method makes no assumptions about the relative displacement and orientation of the pictures. The steps of their algorithm are as follows:

- 1) Segment the image.
- 2) To each region  $i$  assign a set  $V_{i1}, \dots, V_{in}$  where  $V_{ij}$  denotes the value of the  $j$ th feature for region  $i$ . These numbers may describe features such as shape, size, position, spectral values, etc. To each pair of regions, region  $i$  from image 1 and region  $j$  from image 2, define the region to region match rating,  $R_{ij}$ , by:

$$R_{ij} = - \sum_{k=1}^n |V_{ik} - V_{jk}| / W_k S_k$$

where  $W_k$  is a normalization factor for the  $k^{\text{th}}$  feature and  $S_k$  is a measure of importance of the  $k^{\text{th}}$  feature. Larger values of  $R_{ij}$  indicate good matches.

- 3) In this step we attempt to improve the accuracy of the rating  $R_{ij}$  by taking into account adjacency information. To each region  $R_i$  in image 1, assign the region  $R_j$  in image 2 which maximizes  $R_{ij}$ . Let  $N_i$  denote the set of regions in image 1 which are neighbors of region  $R_i$  and have a match in image 2. We say two regions are neighbors if

they have a common boundary point. Let  $N_j$  denote those regions in image 2 which match the regions in  $N_i$ . Let  $N_j'$  denote those regions in  $N_j$  which are neighbors of  $R_j$ . The neighbor feature value (NFV) of  $R_i$  is defined to be the number of elements in  $N_i$  and the NFV for  $R_j$  is the number of elements in  $N_j'$ . Recompute the  $R_{ij}$ 's using this additional feature and assign to each region in image 1 the region in image 2 which matches it best.

This matching algorithm is designed to be invariant under rotation and translation of the images. By omission of size features, the algorithm can be made invariant to scale change. Dissimilar image matching and image to map matching can be handled by this method.

#### 4.2.2 Graph Theoretical Region Matching

This section deals with a method for finding corresponding regions in an image and a map. The procedure [Pavlidis 1977] described takes two lists of regions, one from the image and one from the map, together with a set of possible matches between image and map regions, and computes a region correspondence maximizing the number of regions matched. We now give a precise formulation of the problem.

Assume the map consists of regions  $S_1, \dots, S_m$  and the image consists of regions  $T_1, \dots, T_n$ . Assume to each map region  $S_i$  we assign a set  $H(S_i)$  of image regions. Intuitively the image regions in  $H(S_i)$  have been assigned to  $S_i$  because they may match it on the basis of properties such as size, shape, texture, etc. Our goal is to find a subset  $R \subseteq S$  and a one-to-one function  $g: R \rightarrow T$  satisfying the following properties:

- 1)  $i \neq j \Rightarrow g(S_i) \neq g(S_j)$  for  $S_i, S_j \in R$
- 2)  $\forall S_i \in R \quad g(S_i) \in H(S_i)$
- 3) Given any other pair  $(R', g')$  satisfying condition (1) and (2),  $|R'| \leq |R|$ , where  $| |$  denotes the number of elements in the set.

The computation of an  $R$  and  $g$  in the above problem can be solved using a method from network flow analysis. Before describing the procedure, we discuss the reasons for studying this method. The labelling of regions, in an image based on region descriptors such as ring and wedge data together with structural information, such as the types of regions which can share a common boundary, can be a very time consuming and error prone task. The complexity of labelling algorithms depends upon the number of regions to be labelled and the extent of ambiguity present. The type of region matching presented here can be viewed as a fast procedure for giving a rough match between image and map regions. Both individual region descriptions and inter-region relation-

ships can be included in the matching procedure. The matching obtained using this algorithm can then serve as input to more costly algorithms for matching additional regions and making corrections on existing matches.

We now describe [Bondy 1976] the matching problem in graph-theoretic terms and give a standard method for solving it.

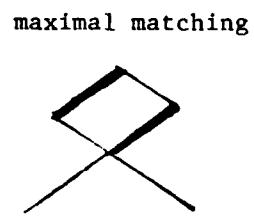
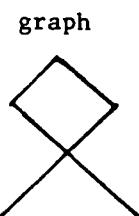
Def. A graph  $G$  is a pair  $(V, E)$  where  $V$  is a finite set  $\{v_1, \dots, v_n\}$  and the  $v_i$  are called nodes or vertices.  $E$  is a set of unordered pairs of elements and its elements are called edges. If  $e_i = (u, v)$ ,  $u, v \in V$  then we may say  $e_i$  joins  $u$  and  $v$ , and  $u$  and  $v$  are endpoints of  $e_i$ , and  $u$  and  $v$  are called adjacent vertices.

Def. A subgraph  $G'$  of a graph  $G = (V, E)$  is a graph  $(V', E')$  such that  $V' \subseteq V$ ,  $E' \subseteq E$ .

Def. An edge with distinct ends is called a link.

Def. Two edges are called adjacent if they share a vertex.

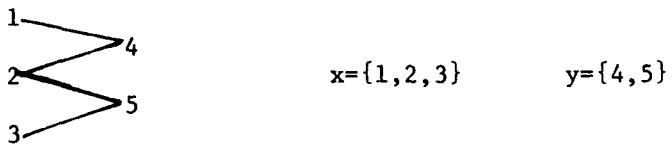
Def. Let  $G = (V, E)$  be a graph. A subset  $M$  of  $E$  is called a matching if  $G$  if its elements are links and no two elements are adjacent in  $G$ .  $M$  is called a maximum matching if  $G$  has no matching  $M'$  with  $|M'| > |M|$ , where  $| |$  denotes the number of edges.



Example 4.1 Maximal matching

Def. A bipartite graph  $G = (V, E)$  is a graph with the property that  $V$  can be

partitioned into two sets  $X$  and  $Y$  such that each edge has one end in  $X$  and one end in  $Y$ .



Example 4.2 Bipartite Graph

We now use the above notation to reformulate our region matching problem. Assume  $S_i$ ,  $T_i$ , and  $H$  are defined as in the beginning of this section. Define a graph  $G = (V, E)$  by  $V = S \cup T$ ,  $E = \{(S_i, T_j) \mid T_j \in H(S_i)\}$ .  $G$  is a bipartite graph with  $V$  partitioned into  $S$  and  $T$ . The solutions,  $(R, g)$ , to our original problem can be put into a one-to-one correspondence with the maximum matchings of the bipartite graph  $G$ . To a solution  $(R, g)$  we correspond the edges given by the graph (in the analysis sense) of  $g$ , i.e.  $(S_i, T_j)$  is an edge iff  $g(S_i) = T_j$ . Conversely, given a maximum matching  $M$  of  $G$ , the corresponding solution  $(R, g)$  is given by:

$g = \{(S_i, T_j) : (S_i, T_j) \in M\}$  (viewing the function  $g$  as a set of ordered pairs).

$R =$  the domain of  $g$ .

We now describe a procedure for finding a maximum matching in a bipartite graph which will have two additional nodes,  $c$  and  $b$ , called a source and a sink respectively, and  $|S| + |T|$  additional edges. In addition each edge will have a direction and a number associated with it. The graph  $G'$  will have an edge joining  $c$  to each node in  $S$ . Each such edge will have weight one and will be directed away from  $c$ . In addition,  $G'$  will have an edge from each node in  $T$  to  $b$ . Each of these nodes will be directed toward  $b$  and will have unit weight. In addition each edge in  $G'$  joining a node in  $S$  and a node in  $T$

will be directed from S to T and will have weight 1.

The maximal matching can now be posed as a network flow problem. A physical analogy is to represent the graph by a set of pipes, one for an edge, where flow occurs only in the specified direction. The vertices in S and T represent connections between pipes. Flow can occur between any two connected pipes subject to the directionality restriction. The numbers assigned to the edges represent the maximum rate of flow through a pipe. Thus, if a pipe, f, with a flow rate n is connected and flowing into a pipe, g, with flow rate m, with  $n > m$ , and with no other pipes connected at this vertex, then f can only support a flow rate of m due to the limit imposed by g. The source c is assumed capable of providing flow to the network at an arbitrarily large rate.

It can be shown that the maximal matching of a bipartite graph can be obtained by finding the flow in  $G'$  maximizing the rate of flow into b. This can be seen intuitively by noting that the restriction to all flow rates being zero or one in  $G'$  allows us to consider a flow of one in an edge joining S and T as the inclusion of this edge in the maximal matching and a flow rate of zero as excluding this edge from the matching. Thus two edges from S to the same node in T would require a flow rate of two to this node, but the flow rate out of such a node is restricted to be one. Hence this situation cannot occur and our matching is one to one. To see that the matching is maximal, note that the flow into b is equal to the sum of the flow rates of the edges going from S to T, which is in turn equal to the number of edges in the matching. We now give a precise description of the network flow problem.

Def. A directed graph, G, on a set  $V = \{v_1, \dots, v_n\}$  is a subset, E, of  $V \times V$  such that for each  $(v_i, v_j) \in E$ ,  $v_i \neq v_j$ . We call the elements of V, the vertices of G and the elements of E the edges of G. We say the  $(v_i, v_j) \in E$  is an

edge from the vertex  $v_i$  to the vertex  $v_j$ . Thus a directed graph may be thought of as an ordinary graph in which each edge is given a direction.

Def. A network  $N$  is a directed graph  $G$  with two distinguished, disjoint, non-empty, subsets  $X$  and  $Y$  of vertices of  $G$  and a non-negative integer valued function  $d$  defined on the edges of  $G$ . The vertices of  $X$  and  $Y$  are called the sources and sinks of  $N$ .

In the notation of our matching problem  $X=\{c\}$  and  $Y=\{b\}$ , i.e.,  $|X|=|Y|=1$ . The set of vertices of  $G$  belonging to neither  $X$  nor  $Y$  are called intermediate vertices of  $G$ , and is denoted by  $I$ . The function  $d$  is called the capacity function of  $N$  and the value of this capacity function on an edge  $e$  is called the capacity of  $e$ . Let  $V$  and  $E$  denote the vertex and edge sets of  $G$ . For any integer-valued function  $f$  on  $E$ , and any  $v \in I$ , define  $f^-(v) = \sum f(e)$  where the sum is taken over all edges in  $E$  of the form  $(v_j, v)$ . Similarly we define  $f^+(v) = \sum f(e)$  where the sum is taken over all edges in  $E$  of the form  $(v, v_i)$ . Thus  $f^+(v)$  represents the flow out of  $v$  and  $f^-(v)$  represents the flow into  $v$ .

Def. A flow in a network  $N$  is an integer-valued function  $f$  defined on  $A$  such that

$$0 \leq f(e) \leq c(e) \quad \text{for all } e \in E$$

and

$$f^-(v) = f^+(v) \quad \text{for all } v \in I.$$

The function  $f$  may be thought of as the assignment of a feasible flow rate through each edge, subject to the restrictions that the flow is limited by the capacity of an edge and that the flows into and out of an intermediate vertex must be equal. From this point onward we restrict ourselves to the case  $X=\{c\}$  and  $Y=\{b\}$ . We define

Def. The value of a flow,  $f$ , on  $n$  is  $\ell(f) = f^+(c)$ .

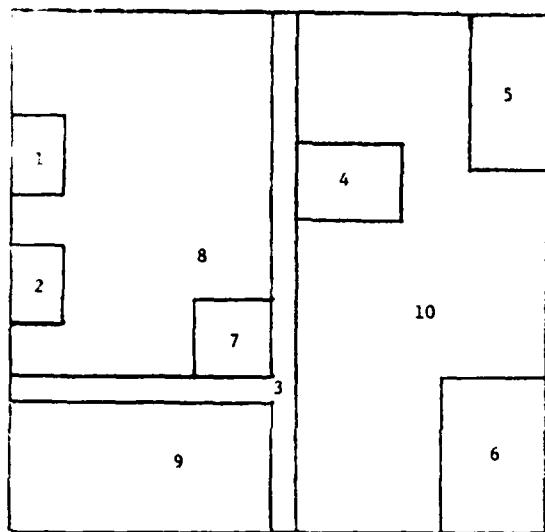
Def. A flow,  $f$ , in  $N$  is called a maximum flow if there is no flow  $g$  in  $N$  such that  $\ell(g) > \ell(f)$ .

Various algorithms [Ford 1962, Nijenhuis 1975] for finding maximum flow in networks have appeared in print. We have used the algorithm given in [Nijenhuis 1975] for our experimentation. For the flow problem arising in maximal matching, with  $n$  regions, i.e.  $|S| + |T| = n$ , and  $m$  edges, the complexity of the algorithm is  $O((n+2)^{1/2} m)$ .

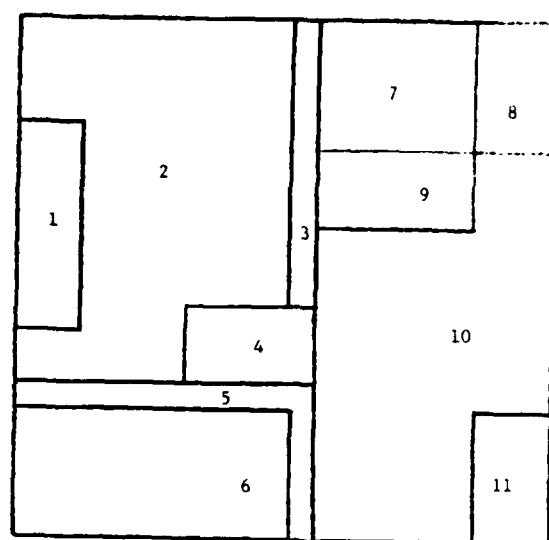
#### 4.2.2.1 Region Matching Experiments

Several region matching experiments were performed to determine the usefulness of various features in performing region matching between two images (or an image and a map) using maximal matching in a bipartite graph. These experiments were performed on synthetic region data. Each artificial image used consisted of a 20x20 grid subdivided into regions but no other information such as gray levels were assumed. Three synthetic images A, B, and C shown in Figure 4.8 were used. Experiments were performed comparing A with B and A with C.

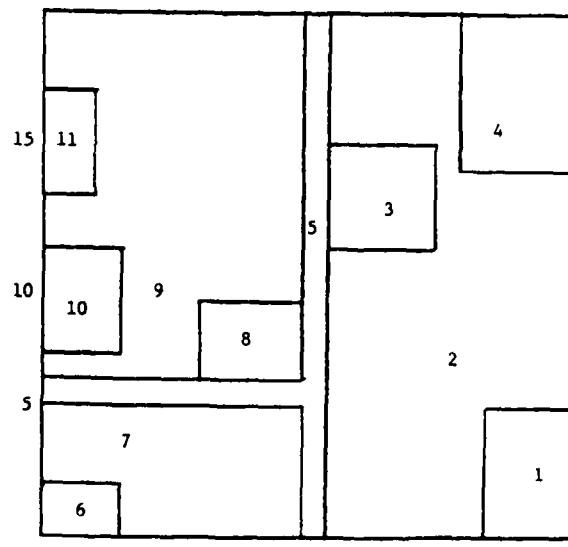
The first experiment was an attempt to determine the adequacy of crude region locations information together with region structure information for region matching. The regions were labelled as shown. Each region  $A_i$  in A was initially matched with all regions in B which had a boundary point within a distance of four from at least one boundary point of the region  $A_i$ . The list of tentative matches is given in Table 4.1. For each region in each image, five region features were computed: area, perimeter, area divided by perimeter, maximum horizontal width of a region and maximum height of a region. The features for the regions in A, B, and C are given in Table 4.2. Not all the features had to be used. If a feature was to be used, the user specified a tolerance  $\alpha$ . If  $f_1$  is the value of a feature for region  $A_i$  and  $f_2$  is the value of the same feature for region  $B_j$  then the two regions fail to match to within the tolerance  $\alpha$  if  $\frac{|f_1-f_2|}{f_1} > \alpha$ . The results of the experiments performed are given in Table 4.3. Of a possible ten correct matches the maximum achieved was four and in this case four incorrect matches were also made. When all features were used with a tolerance of .8, only three correct matches were made, but no incorrect matchings were formed. All other cases had three or fewer correct matches and at least one incorrect match. The graph matches are given in Table 4.4.



(a) Image A



(b) Image B



(c) Image C

Figure 4.8 The artificial images used for region matching experiments.

Table 4.1 Feasible Matching of Image B to Image A

Distance 4 square - The regions can match if their boundaries are no more than 4 squares apart (in the space of distances between curves)

Region A	Region B matches
1	1,2
2	1,2,4,5,6
3	1,2,3,4,5,6,7,9,10
4	2,4,3,7,8,9,10
5	7,8,9,10
6	10,11
7	2,3,4,5,6,9,10
8	1,2,3,4,5,6,7,9,10
9	1,2,3,4,5,6,10
10	2,3,4,5,6,7,8,9,10,11

Table 4.2

## Picture A Features

	Area	Perimeter	Area/Perimeter	Max Hor.	Max Vert.	Rectangle Centers
1	6	10	.6	2	3	(1,5)
2	6	10	.6	2	3	(1,10)
3	30	62	.5	11	20	(5,10)
4	12	14	.8	4	3	(13,14)
5	18	18	1.0	3	6	(19,17)
6	24	20	1.2	4	6	(18,3)
7	9	12	.7	3	3	(9,8)
8	119	56	2.1	10	14	(5,13)
9	50	30	1.7	10	5	(5,3)
10	126	66	1.9	9	20	(15,10)

## Picture B Features

	Area	Perimeter	Area/Perimeter	Max Hor.	Max Vert.	Rectangle Centers
1	16	20	.8	2	8	(1,12)
2	112	52	2.1	10	14	(5,13)
3	11	24	.4	1	11	(9,8)
4	15	16	.9	5	3	(5,3)
5	16	34	.5	11	6	(5,3)
6	50	30	1.7	10	5	(14,18)
7	30	22	1.4	6	5	(19,18)
8	15	16	.9	3	5	(14,14)
9	18	18	1.0	6	3	(16,8)
10	102	48	2.1	9	15	(19,3)
11	15	16	.9	3	5	

## Picture C Features

	Area	Perimeter	Area/Perimeter	Max Hor.	Max Vert.	Rectangle Centers
1	15	16	.9	3	5	(19,3)
2	125	66	1.9	9	20	(16,10)
3	16	16	1.0	4	4	(13,13)
4	24	20	1.2	4	6	(18,17)
5	30	62	.5	11	20	(5,10)
6	6	10	.6	3	2	(2,1)
7	44	30	1.5	10	5	(5,3)
8	12	14	.8	4	3	(8,8)
9	108	58	1.9	10	14	(5,13)
10	12	14	.8	3	4	(2,9)
11	8	12	.7	2	4	(1,15)

Table 4.3 Experiments in Matching Image B to Image A using the vectors 4 square  
Feasible Matchings from Table 4.1.

Feature codes

1. Area of region
2. Perimeter of region
3. Area/Perimeter ratio
4. Maximum horizontal extent of region
5. Maximum vertical extent of region

Trial 1			<u>Matched Regions</u>		Correctness of Match
Features Used	Tolerance	Image A	Image B		
4	.3	2	6	Wrong	
5	.3	3	5	Right	
		6	11	Right	
		10	10	Right	

Trial 2			<u>Matched Regions</u>		Correctness of Match
Features Used	Tolerance	Image A	Image B		
1	.8	3	6	Wrong	
2	.8	4	4	Wrong	
3	.8	6	11	Right	
4	.8	8	7	Wrong	
5	.8	10	10	Right	

Trial 3			<u>Matched Regions</u>		Correctness of Match
Features Used	Tolerance	Image A	Image B		
1	.5	3	5	Right	
2	.5	6	11	Right	
3	.5	10	10	Right	
4	.5				
5	.5				

Trial 4			<u>Matched Regions</u>		Correctness of Match
Features Used	Tolerance	Image A	Image B		
1	.5	3	5	Right	
3	.5	4	4	Wrong	
		6	11	Right	
		10	10	Right	

Trial 5			<u>Matched Regions</u>		Correctness of Match
Features Used	Tolerance	Image A	Image B		
1	.5	3	7	Wrong	
		4	8	Wrong	
		5	9	Wrong	
		6	11	Right	
		7	3	Wrong	
		8	2	Right	
		9	6	Right	
		10	10	Right	

Table 4.4 True matches of Image A Regions with Regions from Image B and C.

Image A	Image B	Image C
1	1	11
2	1	10
3	3,5	5
4	9	3
5	8	4
6	11	1
7	4	8
8	2	9
9	6	7,6
10	10,7	2

In the second experiment, the original criterion for arriving at a tentative match was modified. For each region a center was computed. The center was defined to be the point with coordinates  $(x, y)$  where  $x$  is the average of the leftmost and rightmost horizontal coordinates of the region (truncated to an integer) and  $y$  is the average of the topmost and bottom most  $y$  coordinates of the region (also truncated). Two regions are tentatively matched if their centers are no more than four units apart. Of the ten regions in image B which have at least one tentative match in region A, only eight correct matches can be made. In matching C and A there are eleven possible correct matches but only ten can be determined since we are seeking a one to one matching. In matching A to B the maximal matching, using no features, is given in Figure 4.9. This matching achieves seven matches out of the maximum of eight possible matches under the distances between center restriction. The matching of A to C, using no features, is given in Figure 4.10. In this matching eight of the ten possible correct matches were obtained.

Image B Regions	Possible Image A Regions	Maximal Matching Image A Regions	Correct Matches
1	1, 2	1	x
2	8	8	x
3	none	none	
4	7	7	x
5	none	none	
6	9	9	x
7	none	none	
8	5	5	x
9	4	4	x
10	10	10	x
11	6	6	x

Figure 4.9 Matching of Image A to Image B where possible matches assume the centers of corresponding regions are no more than four units apart.

Image C Regions	Possible Image A Regions	Maximal Matching Image A Regions	Correct Matches
1	6	6	x
2	10	10	x
3	4, 10	4	x
4	5	5	x
5	2, 3	2	
	9	9	
7	9	none	
8	7	7	x
9	8, 3	8	x
10	2, 3	3	
11	1	1	x

Figure 4.10 Matching of Image A to Image C where possible matches assume the centers of corresponding regions are no more than four units apart.

#### 4.2.2.2 Graph Matching Discussion

The evaluation of graph matching algorithms for region labelling is a complicated problem due to the large number of variables involved. Maximum matching in bipartite graphs has serious limitations such as the inability to account for both fragmentation and coalescing of regions. Since the matching is one to one, a region which becomes fragmented can be matched to at most one of its fragments. One possible way to avoid this problem is to redefine the problem by changing the weights on the network graph edges. Suppose we have two images, A and B and that the flow is from the source to image A regions to image B regions to the sink. If each edge from an image A region,  $A_i$  to an image B region,  $B_j$ , is assigned the area of  $A_i$  as a capacity and the edge from  $B_j$  to the sink is assigned the area of  $B_j$  as a capacity then the maximal flow algorithm allows for multiple matchings of regions subject to the constraint that the total area of all regions in A matching with  $B_j$  is no greater than the area of  $B_j$ . This change in the algorithm increases the complexity of the algorithm as well as introducing other complexities in interpretation. An alternate approach to eliminating the effects of region fragmentation is to perform region merging as a preprocessing operation. Selecting adequate merging criteria can be expensive and offsets the advantages of graph matching.

The selection of criteria for the tentative region matches for creating the bipartite graph should be based on extensive analysis of image data. The reliability of region features such as shape, the approximate location of region centers, ring and wedge data, etc. must be determined for a large collection of regions. Hand matching of regions in images taken at different times or matching of images to maps or region data bases can provide information on the variability of these features. If the variation of features is not great then requirements for matching can be made stringent, though this necessitates high quality region feature extraction.

5.0 Verification of Lineal Features

In the process of verifying lineal features, it is assumed that enough information has been extracted from an image so that hypotheses about the remaining image content can be made. In the verification of the existence of a particular lineal feature, the rough location and orientation of the feature is known. That is, if the feature exists at all, the model being used should predict approximately where the feature is with respect to previously detected features, how long it is, what its shape is, etc. For example, if features resembling the two wings of an airplane have been detected, there are at most two places to search for the tail. Finding the hypothesized feature greatly increases the confidence in the model that generated the hypothesis, while failure to detect the predicted feature has just the reverse effect.

Because verification is done with model prediction, focused searches can be performed. Not only is the area of imagery to be searched well-confined, but there are also tight constraints on shape and orientation. Thus faint or hard to detect features can be found more reliably and more efficiently in the top-down mode than in the bottom-up mode.

The technique that has been used views the feature as a set of high gradient points that must be found in the image. The transformation of the image to the model, obtained by running the registration software, is instrumental in the verification process. The inverse of this transformation is used to predict where the undetected feature should exist. The verification software then proceeds to determine whether the hypothesized feature actually exists.

5.1        Technique Used in Verification Process

5.1.1      Profile Search

As an example of the verification process consider the verification of the existence of Sherman Creek in the imagery of Harrisburg, Pennsylvania, as presented in Figure 3.2. It is easy to store the path of the stream as an ordered set of points in some coordinate system, as is done in cartographic data bases. Given a registration of the imagery containing the creek to the cartographic coordinate system, it is easy to transform the points of the mapped feature to the points (pixels) of the image where the feature should be found. Due to noise, distortion, approximation in the registration transformation, and actual change in the stream, it is unlikely that the feature points will be found exactly where they are predicted to be. In an effort to find exactly where the feature point does lie, a technique of "profile searching" is used.

In the profile search, a search for the feature point is made along a "profile" that extends from the predicted point, perpendicular to the direction of the feature, for a distance defined by the user specified tolerance. The search along this profile begins at the predicted point, and proceeds outward in both directions. The search continues in both directions until a "peak point" is found. A peak point is defined as a point that possesses the same gradient direction as the predicted point and has a gradient magnitude that is above the user supplied threshold and is greater than the next point along the profile. This technique will be further clarified through the use of an example.

In the example presented in Figure 5.1, the predicted feature point was at location (7232,4045) in stage coordinates. The window sampling the image is a 32 pixel x 32 pixel array, centered at (16,16). The end points of the profile are dependent on the user supplied tolerance value and the direction of the feature gradient at the predicted point. For this point, the profile extends from (27,20) to (5,12) in pixels. The gradient magnitude is calculated at each of these points using the direction of the gradient at the predicted point. These values are presented on the line with MAG at the left. The user supplied threshold in this case was 4. The search begins at the point (16,16) and proceeds outward in both the direction to the right and to the left. The threshold value is exceeded for the first time at point (13,15) where the gradient magnitude is 27. At this point in the searching process, the left search is at point (19,17) whose magnitude is below threshold. The search continues in both directions. At point (12,14) the gradient magnitude increases to 38 and at point (20,17) the gradient magnitude is still below threshold. At point (11,14) the value of the gradient magnitude decreases, indicating that a peak point has been detected. Since the left search values are still below threshold, the algorithm declares that the actual feature point, in terms of pixel coordinates, lies at point (12,14) instead of point (16,16).

#### 5.1.2 Quality Evaluation

When the verification process has been completed for a specified lineal feature, an evaluation must be made as to how far the detected feature points deviated from their predicted position. A root-mean-

STAGE FOCUSED AT 7232,4045

J,ANGLE,SLOP,K = 1 .393 .414 92

XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12

XTAIL	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5
YTAIL	20	19	19	19	18	18	18	17	17	17	16	16	15	15	15	14	14	14	13	13	13	12	12
DIR	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31
MAG	-5	0	1	2	7	6	4	-1	-1	1	-7	-22	-31	-4	27	38	20	4	1	1	0	0	-1

MAX. GRAD FOR POINT # 1 WAS 38 AT 12,14

Figure 5.1 Example of profile search for a selected point

square value is one measure of the match. Let  $P = P_1, \dots, P_m, \dots, P_n$  be the feature point set in map coordinates and let  $T_\alpha$  be the transformation for registering the imagery to the map coordinate system. Let  $q_m$  be the best detection of point  $P_m$  along the profile passing through  $P_m$ . Then one measure of the verification of the point set  $P$  is

$$D(p) = \left( \left( \sum_{m=1}^n d^2(T_\alpha^{-1}(P_m), q_m) \right) / N \right)^{1/2}$$

where  $d^2(\cdot, \cdot)$  is the squared distance between two image points.

This measure is only taken over those points which have a point of detection on their profile. So in actuality two measures of quality must be used. The first is the  $D(P)$  above and the second is the percentage of unmatched points that are present for that feature.

### 5.1.3 Servoeing

As was stated earlier, due to several factors, it is unlikely that the feature points will be exactly where they are predicted to be. In the experiments run, this was found to be true. In several cases, the points of the feature were consistently removed from the predicted values by some fixed amount. In order to track the actual feature better and to overcome the inaccuracies of the prediction a "servoeing" mechanism was instituted. The servoeing mechanism uses information from previous points to make adjustments to the predicted point. The method used keeps track of the differences between the predicted point and the detected maximum on the profile for as many as the last 5 points. The differences in the X-direction and the Y-direction are averaged and are added to the predicted coordinates to produce a set of "adjusted" point coordinates.

This technique has been implemented and tested. With servoeing, the tracking of the features has greatly improved. The adjusted predicted points are consistently as close or closer to the actual peak points than the original predicted points were. The servoeing also helps to overcome the "warping" effect that is present in most aerial photographs. Because this warping is present, the registration transformation cannot be accurate over the entire picture. Therefore, there are features present in the photograph that may not be accurately verified if only the inverse of the registration transformation is used. The servoeing technique makes the minor adjustments that are needed to overcome these inherent inaccuracies.

## 5.2 Examples from 4621

The verification software was run on several different photographs, the first of which was the photograph called "4621", in Figure 1.1. As can be seen, most of the features of this photograph are straight lines, some very prominent and others not very clear at all.

The registration procedure had been run on a map and image pair using this scene. The resulting transformation was  $(308^\circ, 578, -1397)$ . Using this transformation as a basis, an inverse transformation was developed for use in the verification process. The inverse transformation was  $(52^\circ, -578, 1397)$ . The verification process used this inverse transformation to verify the existence of a set of lineal features in the image. Some of the features verified were quite distinct, while others, such as the feature from point 2 to point 27, were less prominent.

The results obtained from running the verification on this image were extremely encouraging. As can be seen in Table 5.2.1, the percentage of matched points for every one of the lines verified was 100%. The values of the distance function were also small for all of the cases.

As examples of the verification process, output from a run that verified two of the features in 4621 is shown in Figure 5-2. The first feature verified is the one running from point #1 to point #29. This is a prominent road that, as can be seen from the presented results possesses strong gradient values. For each predicted point a profile was constructed and a search was made along the profile for the peak value. The location of this peak value was recorded by the servoeing

Table 5.2.1  
Results of Verification of Features in 4621

Begin Point of Line	End Point of Line	Total # of Points Checked	# of Unmatched Points	% of Matched Points	Distance D(P) in Stagels
1	29	41	0	100%	8.9
2	27	21	0	100%	19.1
1	29	41	0	100%	10.3
8	21	41	0	100%	4.7
3	14	41	0	100%	11.7
16	15	21	0	100%	21.4
12	11	21	0	100%	34.8
3	28	11	0	100%	16.2

```

062 FMC06
:RUN,VER11,6,6
POSITION STAGE TO INITIAL POINT BY HAND AND INPUT COORDINATES.
FORMAT IS XPT, YPT, INT/FREE.
GIVE VALUE FOR DTOL (INT/FREE)
100.
GIVE VALUE FOR DUNIT # AND GRAD THRESHOLD VALUE (INT/FREE)
1.10.
GIVE DEBUG FLAG (VALUE BETW 0 & 30)
3
GIVE REVTHE,REVXS,REVYS,IXCENT,IYCENT,MXCENT,MYCENT
52,-578,-1397,-7500,9500,9500,9500
DTOL,REVTHE,REVXS,REVYS = 100 1.2 -578, -1397.
IXCENT,IYCENT,MXCENT,MYCENT = 7500 9500 9500 9500
GIVE TRANSFORM INFO
308,578,-1397,-9700,31,43,30,30,
0MATCH QUALITY BEFORE VERIFICATION.
THE TA XSHIFT YSHIFT MTCWHT MCHROW/NIMAGE NMCHCOL/NMAP#
308 578 -1397 9700 31/ 43 30/ 30 *****
GIVE EDGE #
1
VERIFYING EDGE # 1
IS THIS A STRAIGHT LINE? INPUT 1 FOR YES, 0 FOR NO
1
GIVE XTAIL,YTAIL (INT/FREE)
4996 7282,
GIVE DELTAX,DELTAY,GRAD. ANG. AND NUMPTS (REAL,REAL,INT,INT)
55.65,113.35,35
55.65,113.35,334,41,
INPUT POINT, X,Y,ANGLE= 4996 7282 334
TRANSFORMED POINT X,Y,G(J) = 5018 4990 31
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 1
ADJUSTED X,Y= 5018 4990
STAGE FOCUSED AT 5018, 4990
J,ANGLE,SLOPE,K = 1 .393 .414 92
XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12
XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
YTAIL 20 19 19 19 18 18 17 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12
DIR 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
MAG 34 12 5 7 2 15 16 4 -12 -40 -61 -13 77 87 59 29 29 29 26 22 21 21 17
MAX. GRAD FOR POINT # 1 WAS 87 AT 14, 15
DIST,TOTPTS,UMCHPT = 17.89 1 0
INPUT POINT, X,Y,ANGLE= 5051 7395 334
TRANSFORMED POINT X,Y,G(J) = 4963 5103 31
Q REC IS-NODE,DELX,DELY,LINK= 2 -16 -8 1
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2
ADJUSTED X,Y= 4955 5099
STAGE FOCUSED AT 4955, 5099
J,ANGLE,SLOPE,K = 1 .393 .414 92
XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12
XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
YTAIL 20 19 19 19 18 18 17 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12
DIR 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
MAG 34 12 5 7 2 15 16 4 -12 -40 -61 -13 77 87 59 29 29 29 26 22 21 21 17
MAX. GRAD FOR POINT # 2 WAS 117 AT 15, 12
DIST,TOTPTS,UMCHPT = 29.20 2 0

MAX. GRAD FOR POINT # 10 WAS 103 AT 15, 15
DIST,TOTPTS,UMCHPT = 357.30 40 0
INPUT POINT, X,Y,ANGLE= 7222 11815 334
TRANSFORMED POINT X,Y,G(J) = 2817 9535 31
Q REC IS-NODE,DELX,DELY,LINK= 1 -8 -8 2
Q REC IS-NODE,DELX,DELY,LINK= 2 -8 -8 3
Q REC IS-NODE,DELX,DELY,LINK= 3 0 0 4
Q REC IS-NODE,DELX,DELY,LINK= 4 -8 -8 5
Q REC IS-NODE,DELX,DELY,LINK= 5 -8 -8 1
ADJUSTED X,Y= 2811 9529
STAGE FOCUSED AT 2811, 9529
J,ANGLE,SLOPE,K = 1 .393 .414 92
XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12
XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
YTAIL 20 19 19 19 18 18 17 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12
DIR 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
MAG 34 12 5 7 2 15 16 4 -12 -40 -61 -13 77 87 59 29 29 29 26 22 21 21 17
MAX. GRAD FOR POINT # 41 WAS 89 AT 17, 16
DIST,TOTPTS,UMCHPT = 365.30 41 0
DIST = 8.9 UMCHPT= 0 OUT OF 41 TOTAL POINTS
WEIGHT = 9109

```

Figure 5.2 Verification results of feature from point #1 to point #29 of 4621, with use of servoeing.

mechanism to be used in adjusting the next predicted point.

In Figure 5.2, it can be seen that the first predicted point was at (5018,4990) of the image. The actual peak was not at the predicted point, which is (16,16) in window coordinates, but at (14,15). In terms of stage coordinates, this means a difference of -16 in the X-direction and -8 in the Y-direction. This is taken into account in the calculation of the next predicted point. As can be seen, the next transformed point in the image is at (4963,5103), but, when the servoing is done the adjusted point becomes (4955,5099). The result of the profile search gives a peak value of 117 at window coordinates (15,15).

Again the actual peak is not at the adjusted predicted point but it is getting closer. For the next 38 points predicted and adjusted for this feature, the actual peak value lies either at (16,16), dead center, or (15,15) just slightly removed. These are excellent results and much better than those obtained without use of the servoing mechanism.

Verification of the same feature, from point #1 to point #29, without the use of servoing is presented in Figure 5.3. In this case, for most of the points the actual peak value was found at either window coordinates (13,15) or (14,15). These values are consistent and close to the center, so it is known that the feature is present but the process does not home in on it. In fact, it was this type of result that led to the creation of the servoing mechanism. Because of transformation approximations, picture warping or other factors, the predicted points in this case are consistently off by a small amount. The servoing mechanism adjusts the tracking by this small amount and homes in on the detected feature.

VERIFYING EDGE # 1  
 IS THIS A STRAIGHT LINE? INPUT 1 FOR ES, 0 FOR NO  
 1  
 GIVE XTAIL,YTAIL (INT/FREE)  
 4996,7282,  
 GIVE DELTAX,DELTAY,GRAD. ANG. AND NUMPTS (REAL,REAL,INT,INT)  
 55,65,113,35,334,41.  
 INPUT POINT, X,Y,ANGLE= 4996 7282 334  
 TRANSFORMED POINT X,Y,G(J) = 5018 4990 31  
 STAGE FOCUSED AT 5018, 4990  
 J,ANGLE,SLOPE,K = 1 .393 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12  
 XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5  
 YTAIL 20 19 19 19 18 18 18 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12  
 DIR 31  
 MAG 52 39 20 9 -1 5 18 11 10 1 -59 -64 7 77 99 55 31 29 29 28 24 25 23  
 MAX. GRAD FOR POINT # 1 WAS 99 AT 13, 15  
 DIST,TOTPTS,UMCHPT = 3.16 1 0  
 INPUT POINT, X,Y,ANGLE= 5051 7395 334  
 TRANSFORMED POINT X,Y,G(J) = 4963 5103 31  
 STAGE FOCUSED AT 4963, 5103  
 J,ANGLE,SLOPE,K = 1 .393 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12  
 XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5  
 YTAIL 20 19 19 19 18 18 18 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12  
 DIR 31  
 MAG -5 -11 2 5 1 -1 -7 12 33 29 -75 \$\$\$ 12 113 139 48 14 10 10 8 10 5 1  
 MAX. GRAD FOR POINT # 2 WAS 139 AT 13, 15  
 DIST,TOTPTS,UMCHPT = 6.32 2 0  
 INPUT POINT, X,Y,ANGLE= 5107 7508 334  
 TRANSFORMED POINT X,Y,G(J) = 4908 5217 31  
 STAGE FOCUSED AT 4908, 5217  
 J,ANGLE,SLOPE,K = 1 .393 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12  
 XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5  
 YTAIL 20 19 19 19 18 18 18 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12  
 DIR 31  
 MAG 15 20 20 5 1 -4 2 24 43 32 -57 \$\$\$ -1 91 129 55 10 -2 11 13 13 11 9  
 MAX. GRAD FOR POINT # 3 WAS 129 AT 13, 15  
 DIST,TOTPTS,UMCHPT = 9.49 3 0

XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5  
 YTAIL 20 19 19 19 18 18 18 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12  
 DIR 31  
 MAG 0 0 -4 -6 -5 0 -3 14 34 29 -75 \$\$\$ -2 100 114 40 4 -1 3 2 4 6 5  
 MAX. GRAD FOR POINT # 4 WAS 114 AT 13, 15  
 DIST,TOTPTS,UMCHPT = 109.10 40 0  
 INPUT POINT, X,Y,ANGLE= 7222 11815 334  
 TRANSFORMED POINT X,Y,G(J) = 2817 9535 31  
 STAGE FOCUSED AT 2817, 9535  
 J,ANGLE,SLOPE,K = 1 .393 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12  
 XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5  
 YTAIL 20 19 19 19 18 18 18 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12  
 DIR 31  
 MAG 1 -1 -2 0 -9 -5 -2 -10 -56 -70 -2 79 90 48 16 10 12 9 7 0 0 -1 1  
 MAX. GRAD FOR POINT # 5 WAS 90 AT 15, 15  
 DIST,TOTPTS,UMCHPT = 110.51 41 0  
 DIST = 2.7 UMCHPT= 0 OUT OF 41 TOTAL POINTS

Figure 5.3 Verification results of feature from point #1 to point #29 of 4621, without use of servoing.

The feature tracked above represents a very prominent roadway. In order to check out the verification process it was also necessary to attempt the verification of features that were not as prominent. One of these less prominent features is the driveway extending from point #2 to point #27. The results of this verification are presented in Figure 54. (Note the peak detection algorithm used in this example is not the most recent scheme. The scheme used here chose the maximum value on the profile regardless of where it was located. It was found that this technique was not as good as the one explained in Section 5.1.3.) Even though this feature is not nearly as prominent as the one discussed above, the procedure was able to verify its existence to a high degree of confidence. Referring to Table 5.2.1, it can be seen that 21 out of 21 predicted points had a match on their profiles.

The results of the experiment run on 4621 were most encouraging. For every feature chosen, a 100% match was obtained with low distance values. However, this experiment only dealt with relatively distinct features that were all straight lines. So, though the results were very satisfying a more difficult set of features was required to continue the testing process. This more challenging set was obtained from the Dreux, France images.

#### Examples from DREUX 13

In an attempt to gain further confidence in the verification

testing was directed to the imagery of

```

GIVE XIAII,YIAIL (INT/FREL)
8730,5494
GIVE DELTAX,DELTAY,GRAD. ANGL. AND NUMPTS (REAL,REAL,INT,INT)
-35.5,-75.6,154,21.
INPUT POINT, X,Y,ANGLE= 8730 5494 154
TRANSFORMED POINT X,Y,G(J) = 8726 6831 15
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 1
ADJUSTED X,Y= 8726 6831
STAGE FOCUSED AT 8726, 6831
J,ANGLE,SLOPE,K = 1 3.534 .414 92
XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12
XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
YTAIL 20 19 19 19 18 18 17 17 17 16 16 15 15 15 14 14 14 14 13 13 13 12 12
DIR 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
MAG -16 -9 -8 -14 -13 -14 14 44 22 -29 -76 -62 -20 -20 -20 -29 -29 -18 -13 -6 -13 -27 -25
MAX. GRAD FOR POINT # 1 WAS 44 AT 20, 17
DIST,TOTPTS,UMCHPT = 32.98 1 0
INPUT POINT, X,Y,ANGLE= 8694 5418 154
TRANSFORMED POINT X,Y,G(J) = 8764 6756 15
Q REC IS-NODE,DELX,DELY,LINK= 2 32 8 1
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2
ADJUSTED X,Y= 8780 6760
STAGE FOCUSED AT 8780, 6760
J,ANGLE,SLOPE,K = 1 3.534 .414 92
XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12
XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
YTAIL 20 19 19 19 18 18 17 17 17 16 16 15 15 15 14 14 14 14 13 13 13 12 12
DIR 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
MAG 3 6 12 18 15 9 0 6 33 32 -14 -45 -17 9 15 -13 -28 -25 -13 3 5 -4 -8
MAX. GRAD FOR POINT # 2 WAS 33 AT 19, 17
DIST,TOTPTS,UMCHPT = 58.28 2 0
INPUT POINT, X,Y,ANGLE= 8659 5342 154
TRANSFORMED POINT X,Y,G(J) = 8802 6682 15
Q REC IS-NODE,DELX,DELY,LINK= 3 24 8 1
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2
Q REC IS-NODE,DELX,DELY,LINK= 2 32 8 3
ADJUSTED X,Y= 8820 6687
STAGE FOCUSED AT 8820, 6687
J,ANGLE,SLOPE,K = 1 3.534 .414 92
XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12
XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
YTAIL 20 19 19 19 18 18 17 17 17 16 16 15 15 15 14 14 14 14 13 13 13 12 12
DIR 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
MAG 13 12 12 11 1 -3 -5 -3 27 37 5 -43 -50 -38 -18 -4 16 32 18 4 -12 -17 -14
MAX. GRAD FOR POINT # 3 WAS 37 AT 18, 17
DIST,TOTPTS,UMCHPT = 76.17 3 0
INPUT POINT, X,Y,ANGLE= 8623 5267 154
TRANSFORMED POINT X,Y,G(J) = 8839 6607 15
Q REC IS-NODE,DELX,DELY,LINK= 4 16 8 1
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2
Q REC IS-NODE,DELX,DELY,LINK= 2 32 8 3
Q REC IS-NODE,DELX,DELY,LINK= 3 24 8 4
ADJUSTED X,Y= 8857 6613
STAGE FOCUSED AT 8857, 6613
J,ANGLE,SLOPE,K = 1 3.534 .414 92
XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12
XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
YTAIL 20 19 19 19 18 18 17 17 17 16 16 15 15 15 14 14 14 14 13 13 13 12 12
DIR 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
MAG 7 12 9 4 8 8 5 -12 6 26 12 -29 -46 -27 -10 -7 -14 -12 -16 -17 -9 6 15
MAX. GRAD FOR POINT # 4 WAS 26 AT 18, 17
DIST,TOTPTS,UMCHPT = 94.06 4 0
INPUT POINT, X,Y,ANGLE= 8588 5191 154
TRANSFORMED POINT X,Y,G(J) = 8877 6533 15
Q REC IS-NODE,DELX,DELY,LINK= 5 16 8 1
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2
Q REC IS-NODE,DELX,DELY,LINK= 2 32 8 3
Q REC IS-NODE,DELX,DELY,LINK= 3 24 8 4

```

Figure 5.4 Results of verification of driveway running from point #2 to point #27 of 4621, with servoeing (part 1 of 2).

Q REC 15-NODE,DELX,DELY,LINK= 4 16 8 5  
 ADJUSTED X,Y= 8894 6539  
 STAGE FOCUSED AT 8894, 6539  
 J,ANGLE,SLOPE,K = 1 3.534 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12  
 XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5  
 YTAIL 20 19 19 19 18 18 18 17 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12  
 DIR 15  
 MAG 14 16 8 4 6 12 6 -18 -11 11 13 -23 -53 -36 -4 15 7 -4 -13 -11 -3 -2 -4  
 MAX. GRAD FOR POINT # 5 WAS 16 AT 26, 19 (under new peak detection (17,16) would have been chosen)  
 DIST,TOTPTS,UMCHPT = 177.58 5 0  
 INPUT POINT, X,Y,ANGLE= 8552 5116 154  
 TRANSFORMED POINT X,Y,G(J) = 8914 6458 15  
 Q REC IS-NODE,DELX,DELY,LINK= 1 80 24 2  
 Q REC IS-NODE,DELX,DELY,LINK= 2 32 8 3  
 Q REC IS-NODE,DELX,DELY,LINK= 3 24 8 4  
 Q REC IS-NODE,DELX,DELY,LINK= 4 16 8 5  
 Q REC IS-NODE,DELX,DELY,LINK= 5 16 8 1  
 ADJUSTED X,Y= 8947 6469  
 STAGE FOCUSED AT 8947, 6469  
 J,ANGLE,SLOPE,K = 1 3.534 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12  
 XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5  
 YTAIL 20 19 19 19 18 18 18 17 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12  
 DIR 15  
 MAG 8 6 16 19 4 -3 0 16 10 -9 -6 14 9 -30 -60 -36 -12 2 -4 -3 -6 -5 -3  
 MAX. GRAD FOR POINT # 6 WAS 19 AT 24, 19 (UNDER NEW PEAK DETECTION (16,16) would have been chosen)  
 DIST,TOTPTS,UMCHPT = 245.93 6 0  
 INPUT POINT, X,Y,ANGLE= 8517 5040 154  
 TRANSFORMED POINT X,Y,G(J) = 8952 6384 15  
 Q REC IS-NODE,DELX,DELY,LINK= 2 64 24 3  
 Q REC IS-NODE,DELX,DELY,LINK= 3 24 8 4  
 Q REC IS-NODE,DELX,DELY,LINK= 4 16 8 5  
 Q REC IS-NODE,DELX,DELY,LINK= 5 16 8 1  
 Q REC IS-NODE,DELX,DELY,LINK= 1 80 24 2  
 ADJUSTED X,Y= 8992 6398  
 STAGE FOCUSED AT 8992, 6398  
 J,ANGLE,SLOPE,K = 1 3.534 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12  
 XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5  
 YTAIL 20 19 19 19 18 18 18 17 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12  
 DIR 15  
 MAG -2 8 13 15 5 -1 -8 -19 -11 1 9 19 48 31 -15 -72 -59 -26 -1 0 2 -3 -2  
 MAX. GRAD FOR POINT # 7 WAS 48 AT 15, 15  
 DIST,TOTPTS,UMCHPT = 257.25 7 0

MAX. GRAD FOR POINT # 20 WAS 39 AT 17, 16  
 DIST,TOTPTS,UMCHPT = 384.21 20 0  
 INPUT POINT, X,Y,ANGLE= 8020 3982 154  
 TRANSFORMED POINT X,Y,G(J) = 9480 5341 15  
 Q REC IS-NODE,DELX,DELY,LINK= 1 8 0 0 2  
 Q REC IS-NODE,DELX,DELY,LINK= 2 8 0 0 3  
 Q REC IS-NODE,DELX,DELY,LINK= 3 8 0 0 4  
 Q REC IS-NODE,DELX,DELY,LINK= 4 8 0 0 5  
 Q REC IS-NODE,DELX,DELY,LINK= 5 8 0 0 1  
 ADJUSTED X,Y= 9488 5341  
 STAGE FOCUSED AT 9488, 5341  
 J,ANGLE,SLOPE,K = 1 3.534 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12  
 XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5  
 YTAIL 20 19 19 19 18 18 18 17 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12  
 DIR 15  
 MAG 8 15 9 -6 -24 -21 -5 19 37 51 19 -15 -67 -69 -34 26 12 -17 -43 -27 -2 15 21  
 MAX. GRAD FOR POINT # 21 WAS 51 AT 18, 17  
 DIST,TOTPTS,UMCHPT = 402.10 21 0  
 DIST = 19.1 UMCHPT= 0 OUT OF 21 TOTAL POINTS

Figure 5.4 Results of verification of driveway running from point #2 to point #27 of 4621, with servoing (part 2 of 2).

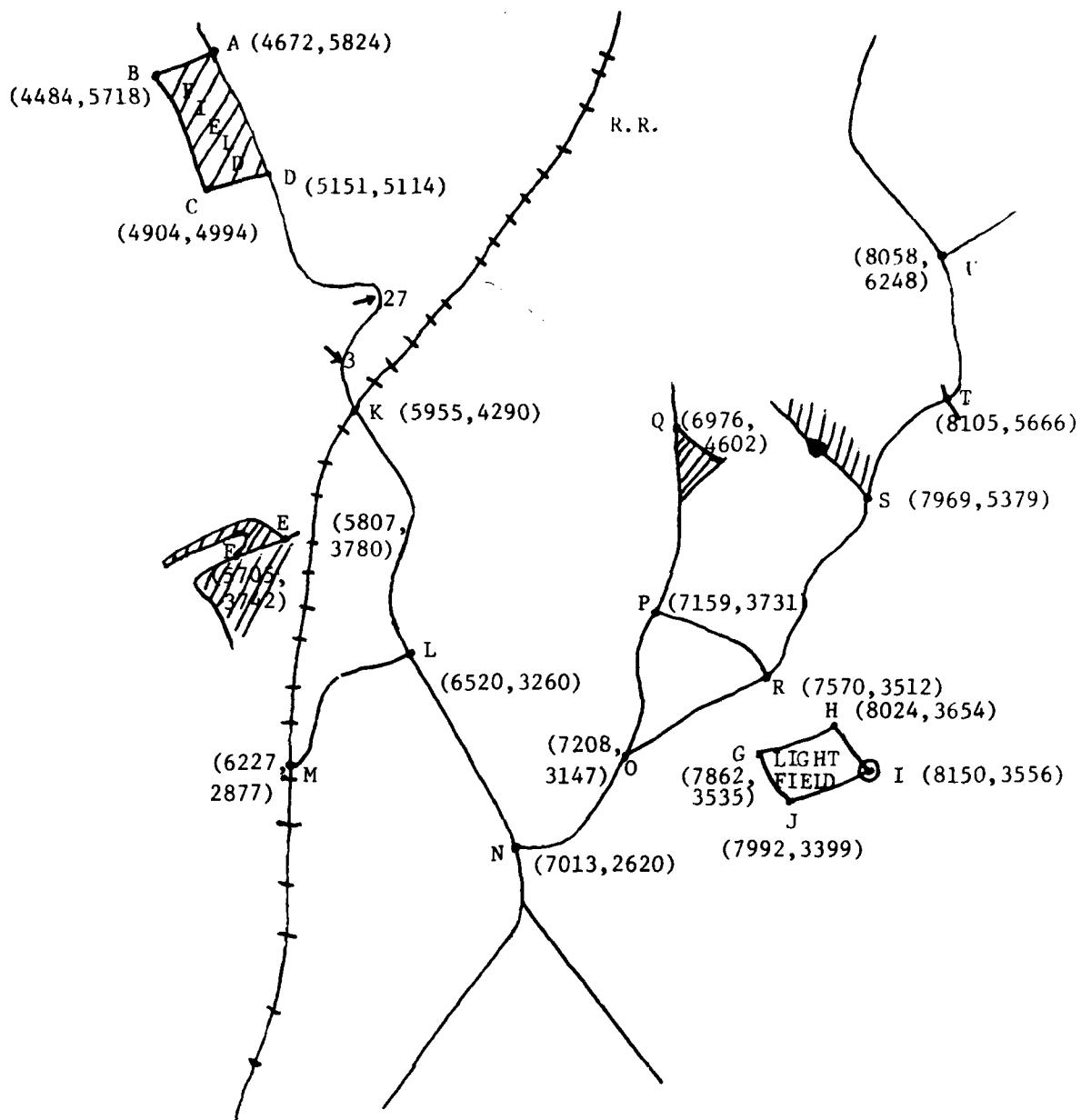


Table 5.3.1  
Results of Verification of features in Dreux 13

Begin Point of Line	End Point of Line	Total # of Points Checked	# of Unmatched Points	% of Matched Points	Distance D(P)
K	D	59	0	100%	3.7
N	Q	87	1	98.9%	7.7
U	O	97	1	98.9%	7.5
E	F	31	5	86.1%	12.2
P	R	19	5	78.7%	26.0
M	L	27	15	44.4%	9.7
Q	T	25	7	72%	42.0

(non-existent lineal)

GIVE FILE NAME OF INPUT FILE = (3A2)

CTDRKD

INPUT POINT, X,Y,ANGLE= 11078 8546 39  
TRANSFORMED POINT X,Y,G(J) = 6279 7768 31  
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 1  
ADJUSTED X,Y= 6279 7768  
STAGE FOCUSED AT 6279, 7768  
J,ANGLE,SLOPE,K = 1 .393 .414 92  
XTAIL,YTAIL,XHEAD,YHEAD = 5 12 27 20  
XTAIL 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27  
YTAIL 12 13 13 13 14 14 15 15 15 16 16 17 17 17 17 18 19 19 19 19 19 19 19  
DIR 31  
MAG -2 0 0 0 -1 0 1 2 7 17 15 -5 -16 -17 -5 -1 0 1 0 -1 2 -1  
MAX. GRAD FOR POINT # 1 WAS 17 AT 15, 16  
DIST,TOTPTS,UMCHPT = 8.00 1 0  
INPUT POINT, X,Y,ANGLE= 11852 8528 43  
TRANSFORMED POINT X,Y,G(J) = 6264 7807 31  
Q REC IS-NODE,DELX,DELY,LINK= 2 -8 0 1  
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2  
ADJUSTED X,Y= 6260 7807  
STAGE FOCUSED AT 6260, 7807  
J,ANGLE,SLOPE,K = 1 .393 .414 92  
XTAIL,YTAIL,XHEAD,YHEAD = 5 12 27 20  
XTAIL 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27  
YTAIL 12 13 13 13 14 14 15 15 15 16 16 17 17 17 17 18 19 19 19 19 19 19 20  
DIR 31  
MAG -1 0 0 0 -1 0 1 0 -2 5 26 28 -5 -23 -24 -5 2 3 1 -3 -5 -4 -2  
MAX. GRAD FOR POINT # 2 WAS 28 AT 16, 16  
DIST,TOTPTS,UMCHPT = 8.00 2 0  
INPUT POINT, X,Y,ANGLE= 11828 8600 42  
TRANSFORMED POINT X,Y,G(J) = 6248 7835 31  
Q REC IS-NODE,DELX,DELY,LINK= 3 0 0 1  
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2  
Q REC IS-NODE,DELX,DELY,LINK= 2 -8 0 3  
ADJUSTED X,Y= 6246 7835  
STAGE FOCUSED AT 6246, 7835  
J,ANGLE,SLOPE,K = 1 .393 .414 92  
XTAIL,YTAIL,XHEAD,YHEAD = 5 12 27 20  
XTAIL 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27  
YTAIL 12 13 13 13 14 14 15 15 15 16 16 17 17 17 17 18 19 19 19 19 19 19 20  
DIR 31  
MAG -2 0 0 0 0 1 0 -3 3 31 38 2 -28 -37 -15 1 4 3 2 1 0 -1  
MAX. GRAD FOR POINT # 3 WAS 38 AT 15, 16  
DIST,TOTPTS,UMCHPT = 8.00 3 0  
INPUT POINT, X,Y,ANGLE= 11798 8638 41  
TRANSFORMED POINT X,Y,G(J) = 6231 7880 31  
Q REC IS-NODE,DELX,DELY,LINK= 4 0 0 1  
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2  
Q REC IS-NODE,DELX,DELY,LINK= 2 -8 0 3  
Q REC IS-NODE,DELX,DELY,LINK= 3 0 0 4  
ADJUSTED X,Y= 6229 7880  
STAGE FOCUSED AT 6229, 7880  
J,ANGLE,SLOPE,K = 1 .393 .414 92  
XTAIL,YTAIL,XHEAD,YHEAD = 5 12 27 20  
XTAIL 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27  
YTAIL 12 13 13 13 14 14 15 15 15 16 16 17 17 17 17 18 19 19 19 19 19 20  
DIR 31  
MAG -1 0 1 1 0 0 1 -1 -1 14 53 51 -14 -49 -47 -11 1 1 2 2 2 0 -1  
MAX. GRAD FOR POINT # 4 WAS 53 AT 15, 16  
DIST,TOTPTS,UMCHPT = 16.00 4 0  
INPUT POINT, X,Y,ANGLE= 11768 8668 43  
TRANSFORMED POINT X,Y,G(J) = 6212 7918 31  
Q REC IS-NODE,DELX,DELY,LINK= 5 -8 0 1  
Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2  
Q REC IS-NODE,DELX,DELY,LINK= 2 -8 0 3  
Q REC IS-NODE,DELX,DELY,LINK= 3 0 0 4

Figure 5.6

Results of verification of feature from point K to point D of DREUX 13, with servoeing.  
(part 1 of 2).

DIST,10TPPTS,UMCHPT = 203.51 55 0  
 INPUT POINT, X,Y,ANGLE= 10420 10082 39  
 TRANSFORMED POINT X,Y,G(J) = 5368 9680 31  
 Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2  
 Q REC IS-NODE,DELX,DELY,LINK= 2 1 0 3  
 Q REC IS-NODE,DELX,DELY,LINK= 3 1 0 4  
 Q REC IS-NODE,DELX,DELY,LINK= 4 1 0 5  
 Q REC IS-NODE,DELX,DELY,LINK= 5 -7 0 1  
 ADJUSTED X,Y= 5368 9680  
 STAGE FOCUSED AT 5368, 9680  
 J,ANGLE,SLOPE,K = 1 .393 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 5 12 27 20  
 XTAIL 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27  
 YTAIL 12 13 13 13 14 14 14 15 15 15 16 16 17 17 17 18 18 19 19 19 20 20  
 DIR 31  
 MAG 0 3 2 5 5 2 3 -7 -10 4 31 28 -28 -45 -35 1 5 4 3 0 -1 0 0  
 MAX. GRAD FOR POINT # 56 WAS 31 AT 15, 16  
 DIST,10TPPTS,UMCHPT = 210.51 56 0  
 INPUT POINT, X,Y,ANGLE= 10378 10130 38  
 TRANSFORMED POINT X,Y,G(J) = 5343 9739 31  
 Q REC IS-NODE,DELX,DELY,LINK= 2 -8 0 3  
 Q REC IS-NODE,DELX,DELY,LINK= 3 1 0 4  
 Q REC IS-NODE,DELX,DELY,LINK= 4 1 0 5  
 Q REC IS-NODE,DELX,DELY,LINK= 5 -7 0 1  
 Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2  
 ADJUSTED X,Y= 5341 9739  
 STAGE FOCUSED AT 5341, 9739  
 J,ANGLE,SLOPE,K = 1 .393 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 5 12 27 20  
 XTAIL 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27  
 YTAIL 12 13 13 13 14 14 14 15 15 15 16 16 17 17 17 18 18 18 19 19 19 20 20  
 DIR 31  
 MAG -2 -1 -4 -3 -4 -5 -3 -2 -12 -6 22 39 -7 -41 -43 -9 7 4 1 0 -1 -3 -2  
 MAX. GRAD FOR POINT # 57 WAS 39 AT 16, 16  
 DIST,10TPPTS,UMCHPT = 210.51 57 0  
 INPUT POINT, X,Y,ANGLE= 10344 10178 41  
 TRANSFORMED POINT X,Y,G(J) = 5325 9795 31  
 Q REC IS-NODE,DELX,DELY,LINK= 3 -2 0 4  
 Q REC IS-NODE,DELX,DELY,LINK= 4 1 0 5  
 Q REC IS-NODE,DELX,DELY,LINK= 5 -7 0 1  
 Q REC IS-NODE,DELX,DELY,LINK= 1 0 0 2  
 Q REC IS-NODE,DELX,DELY,LINK= 2 -8 0 3  
 ADJUSTED X,Y= 5322 9795  
 STAGE FOCUSED AT 5322, 9795  
 J,ANGLE,SLOPE,K = 1 .393 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 5 12 27 20  
 XTAIL 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27  
 YTAIL 12 13 13 13 14 14 14 15 15 15 16 16 17 17 17 18 18 18 19 19 19 20 20  
 DIR 31  
 MAG -2 3 0 -4 -7 -5 -5 -12 -8 11 38 18 -38 -43 -21 5 3 -2 2 2 0 -3 -3 -3  
 MAX. GRAD FOR POINT # 58 WAS 38 AT 15, 16  
 DIST,10TPPTS,UMCHPT = 210.51 58 0  
 INPUT POINT, X,Y,ANGLE= 10288 10232 46  
 TRANSFORMED POINT X,Y,G(J) = 5289 9863 31  
 Q REC IS-NODE,DELX,DELY,LINK= 4 -11 0 5  
 Q REC IS-NODE,DELX,DELY,LINK= 5 -7 0 1  
 Q REC IS-NODE,DELX,DELY,LINK= 1 8 0 2  
 Q REC IS-NODE,DELX,DELY,LINK= 2 -8 0 3  
 Q REC IS-NODE,DELX,DELY,LINK= 3 -2 0 4  
 ADJUSTED X,Y= 5284 9863  
 STAGE FOCUSED AT 5284, 9863  
 J,ANGLE,SLOPE,K = 1 .393 .414 92  
 XTAIL,YTAIL,XHEAD,YHEAD = 5 12 27 20  
 XTAIL 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27  
 YTAIL 12 13 13 13 14 14 14 15 15 15 16 16 17 17 17 18 18 18 19 19 19 20 20  
 DIR 31  
 MAG -3 -7 -7 -7 -8 -10 -12 -7 -8 2 43 53 2 -40 -50 -11 5 2 -3 -4 -3 0 1  
 MAX. GRAD FOR POINT # 59 WAS 53 AT 16, 16  
 DIST,10TPPTS,UMCHPT = 218.51 59 0  
 DIST = 3.7 UMCHPT= 0 OUT OF 59 TOTAL POINTS  
 WEIGHT = 28863  
 GIVE EDGE #  
 -1  
 GIVE REVTHE,REVXS,REVYS,IXCENT,IYCENT,MXCENT,MYCENT  
 -1

Figure 5.6 Results of verification of feature from point K to point D of DREUX 13, with servoeing. (part 2 of 2)

are quite high. The two sides of the road are evident in the profile by the high positive values adjacent to the low negative values. The verifying of the line proceeds quite accurately even around the sharp curve in the road. The detected peak is always no further than 2 pixels off the center, thus resulting in the extremely low value for  $D(P)$ .

The results obtained were not always as good as in the above case. The feature extending from point P to point R, is a very faint road that runs into a group of buildings near point R. The results of the verification of this road are presented in Figure 5.7. The magnitude values for the gradient in the area of this line are low. For many of the profiles the threshold value is never exceeded, therefore resulting in unmatched points. However, on closer examination there are definitive observable peaks in the profiles but, the peak values are lower than the user supplied threshold of 4. Also when predicted points close to point R are examined, large gradient values appear at a significant distance from the predicted point. These large values are the result of the profile passing through the area where the buildings are. So though the results are not as good as with some of the other features, a human examination of the area can explain some of the reasons for this.

An experiment was also run to try to verify the existence of a non-existent feature running from point Q to point T. To accomplish this, a set of points was taken along a fictitious feature starting at point Q and ending at point T. This set of points was then presented to the verification procedure. Though the procedure was able to match

```

IS THIS A STRAIGHT LINE? INPUT 1 FOR YES, 0 FOR NO
0
IS INPUT IN A FILE? (1 FOR YES, 0 FOR NO)
1
GIVE FILE NAME OF INPUT FILE - (3A2)
CINNPR
INPUT POINT, X,Y,ANGLE= 14296 7422 268
TRANSFORMED POINT X,Y,G(J) = 8240 5948 11
Q REC IS-NODE,DEIX,DELY,LINK= 1 0 0 1
ADJUSTED X,Y= 8240 5948
STAGE FOCUSED AT 8240, 5948
J,ANGLE,SLOPE,K = 1 4.319 2.412 38
XTAIL,YTAIL,XHEAD,YHEAD = 20 27 12 5
XTAIL 20 19 19 18 18 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12
YTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
DIR 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
MAG 1 1 2 4 3 2 2 1 1 1 -2 2 -1 1 -1 1 -1 1 -1 1 -3 -2 0 -1 -3
MAX. GRAD FOR POINT # 1 WAS 4 AT 19, 24
DIST,TOTPTS,UMCHPT = 68.35 1 0
INPUT POINT, X,Y,ANGLE= 14388 7402 257
TRANSFORMED POINT X,Y,G(J) = 8321 5901 11
Q REC IS-NODE,DEIX,DELY,LINK= 2 24 64 1
Q REC IS-NODE,DEIX,DELY,LINK= 1 0 0 2
ADJUSTED X,Y= 8333 5933
STAGE FOCUSED AT 8333, 5933
J,ANGLE,SLOPE,K = 1 4.319 2.412 38
XTAIL,YTAIL,XHEAD,YHEAD = 20 27 12 5
XTAIL 20 19 19 18 18 17 17 16 16 15 15 15 15 14 14 14 13 13 13 12 12
YTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
DIR 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
MAG -1 0 0 0 1 1 0 0 0 2 3 3 0 0 -2 -1 2 5 3 -5 -6 -4 1
MAX. GRAD FOR POINT # 2 WAS 5 AT 14, 10
DIST,TOTPTS,UMCHPT = 118.95 2 0

```



```

TRANSFORMED POINT X,Y,G(J) = 8863 5361 11
Q REC IS-NODE,DEIX,DELY,LINK= 3 25 -5 4
Q REC IS-NODE,DEIX,DELY,LINK= 4 1 -8 5
Q REC IS-NODE,DEIX,DELY,LINK= 5 -13 -21 1
Q REC IS-NODE,DEIX,DELY,LINK= 1 -14 -22 2
Q REC IS-NODE,DEIX,DELY,LINK= 2 33 0 3
ADJUSTED X,Y= 8869 5350
STAGE FOCUSED AT 8869, 5350
J,ANGLE,SLOPE,K = 1 3.534 .414 92
XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12
XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
YTAIL 20 19 19 19 18 18 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12
DIR 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
MAG 3 -4 -5 -2 -7 -11 -11 -3 4 10 13 2 -13 -13 -7 1 1 0 -2 -3 -2 -1 -1 -1
MAX. GRAD FOR POINT # 18 WAS 13 AT 17, 16
DIST,TOTPTS,UMCHPT = 306.40 18 5
INPUT POINT, X,Y,ANGLE= 15112 7010 222
TRANSFORMED POINT X,Y,G(J) = 8888 5304 15
Q REC IS-NODE,DEIX,DELY,LINK= 4 14 -11 5
Q REC IS-NODE,DEIX,DELY,LINK= 5 -13 -21 1
Q REC IS-NODE,DEIX,DELY,LINK= 1 -14 -22 2
Q REC IS-NODE,DEIX,DELY,LINK= 2 33 0 3
Q REC IS-NODE,DEIX,DELY,LINK= 3 25 -5 4
ADJUSTED X,Y= 8897 5293
STAGE FOCUSED AT 8897, 5293
J,ANGLE,SLOPE,K = 1 3.534 .414 92
XTAIL,YTAIL,XHEAD,YHEAD = 27 20 5 12
XTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
YTAIL 20 19 19 19 18 18 17 17 16 16 15 15 15 14 14 14 13 13 13 12 12
DIR 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
MAG 45 30 14 11 35 34 30 6 -1 -7 -4 3 -19 -44 -45 -12 -2 0 -1 0 -1 -1 0
MAX. GRAD FOR POINT # 19 WAS 35 AT 23, 18
DIST,TOTPTS,UMCHPT = 364.64 19 5
DIST = 26.0 UMCCHPT= 5 OUT OF 19 TOTAL POINTS

```

Figure 5.7 Results of verification of feature from point P to point R of DREUX 13, with servoing.

72% of the points, the value for  $D(P)$  is much greater than those obtained for any of the other features. Upon examining the output produced (see Figure 5.8), it is noted that there is no recognizable pattern to the location of the peaks. This leads to the high value of  $D(P)$  and decreases ones confidence that the features exists in the image, and in fact it does not.

The results obtained in running the verification software on the DREUX 13 image have buoyed our confidence in its ability to verify features in an image. It demonstrated the ability to follow highly curved lines, find both distinct and not so distinct features and indicate failure when an attempt is made to track a feature that is not present.

GIVE FILE NAME OF INPUT FILE - (3A2)

CTDRQ1

```
INPUT POINT, X,Y,ANGLE= 13896 9182 282
TRANSFORMED POINT X,Y,G(J) = 8404 7745 9
Q REC IS-NODE,DFLX,DELY,LINK= 1 0 0 1
ADJUSTED X,Y= 8404 7745
STAGE FOCUSED AT 8404, 7745
J,ANGLE,SLOPE,K = 1 4.7122803.561 0
XTAIL,YTAIL,XHEAD,YHEAD = 16 25 16 7
XTAIL 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
YTAIL 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7
DIR 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
MAG -1 -1 -1 0 1 1 0 -1 -1 1 2 2 1 0 -1 0 1 2 1
MAX. GRAD FOR POINT # 1 WAS 1 AT 16, 7
DIST,TOTPTS,UMCHPT = .00 1 1
```



```
-- TNPNTY PTNPT, X,Y,ANGLE= 15954 11034 304
TRANSFORMED POINT X,Y,G(J) = 10934 8870 7
Q REC IS-NODE,DEIX,DELY,LINK= 1 46 -93 2
Q REC IS-NODE,DEIX,DFLY,LINK= 2 16 -60 3
Q REC IS-NODE,DFLX,DELY,LINK= 3 -15 -32 4
Q REC IS-NODE,DFLX,DELY,LINK= 4 58 -106 5
Q REC IS-NODE,DFLX,DELY,LINK= 5 67 -118 1
ADJUSTED X,Y= 10968 8789
STAGE FOCUSED AT 10968, 8789
J,ANGLE,SLOPE,K = 1 5.105 -2.417 38
XTAIL,YTAIL,XHEAD,YHEAD = 12 27 20 5
XTAIL 12 13 13 13 14 14 14 15 15 15 16 16 17 17 17 17 18 18 18 19 19 19 20 20
YTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
DIR 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
MAG 1 1 2 1 0 1 2 2 1 0 1 2 2 1 0 1 2 2 1 0 0 1 2 21 45 56 12 -29
MAX. GRAD FOR POINT # 23 WAS 56 AT 19, 7
DIST,TOTPTS,UMCHPT = 676.02 23 7
INPUT POINT, X,Y,ANGLE= 16068 11124 311
TRANSFORMED POINT X,Y,G(J) = 11070 8920 7
Q REC IS-NODE,DEIX,DELY,LINK= 2 58 -153 3
Q REC IS-NODE,DEIX,DELY,LINK= 3 -15 -32 4
Q REC IS-NODE,DEIX,DELY,LINK= 4 58 -106 5
Q REC IS-NODE,DEIX,DELY,LINK= 5 67 -118 1
Q REC IS-NODE,DEIX,DELY,LINK= 1 46 -93 2
ADJUSTED X,Y= 11112 8820
STAGE FOCUSED AT 11112, 8820
J,ANGLE,SLOPE,K = 1 5.105 -2.417 38
XTAIL,YTAIL,XHEAD,YHEAD = 12 27 20 5
XTAIL 12 13 13 13 14 14 14 15 15 15 16 16 17 17 17 17 18 18 18 19 19 19 20 20
YTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
DIR 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
MAG 3 1 1 0 0 2 5 4 2 6 25 42 24 -12 -43 -28 -7 5 4 1 -1 0
MAX. GRAD FOR POINT # 24 WAS 42 AT 17, 15
DIST,TOTPTS,UMCHPT = 687.34 24 7
INPUT POINT, X,Y,ANGLE= 16224 11270 313
TRANSFORMED POINT X,Y,G(J) = 11264 9011 7
Q REC IS-NODE,DEIX,DELY,LINK= 3 50 -108 4
Q REC IS-NODE,DEIX,DELY,LINK= 4 58 -106 5
Q REC IS-NODE,DEIX,DELY,LINK= 5 67 -118 1
Q REC IS-NODE,DEIX,DELY,LINK= 1 46 -93 2
Q REC IS-NODE,DEIX,DELY,LINK= 2 58 -153 3
ADJUSTED X,Y= 11319 8896
STAGE FOCUSED AT 11319, 8896
J,ANGLE,SLOPE,K = 1 5.105 -2.417 38
XTAIL,YTAIL,XHEAD,YHEAD = 12 27 20 5
XTAIL 12 13 13 13 14 14 14 15 15 15 16 16 17 17 17 17 18 18 18 19 19 19 20 20
YTAIL 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
DIR 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
MAG 19 5 -8 -17 -16 -10 -2 2 0 -3 -5 -1 2 2 1 -1 -1 1 2 4 2 -1 1
MAX. GRAD FOR POINT # 25 WAS 4 AT 19, 8
DIST,TOTPTS,UMCHPT = 755.69 25 7
DIST = 42.0 UMCPT= 7 OUT OF 25 TOTAL POINTS
WEIGHT = 13196
```

Figure 5.8 Results of attempted verification of non-existent lineal from point Q to point T.

Discussion and conclusions on verification

The work on verification should be regarded as pilot work and too weak to support firm conclusions. The question of evaluation appears to be difficult at two distinct levels. First of all, how do we actually score the degree of match between a stored lineal track and a set of observed points from the image? How should the extracted score be interpreted? Ultimately the answer to this question must come from an actual purposed application. The current experiments reported only the number of points matched and the popular RMS distance between observed and predicted points. At the second level, how is the performance of a verifier to be evaluated over a set of different lineals? Clearly the answer is dependent on the scoring at the first level. How well did the verifier perform on river I and how well did it perform on the 7 images of France?

More research towards solving these questions needs to be done. At present only subjective or qualitative statements can be made about the past work. In general it can be said that the verification procedure performed consistently well (i.e. produced good numbers with little variance) on features which were distinct to the human observer. These were major roads and clear field boundaries where gradient values were consistently above threshold and agreed in location with the human digitization. Thus verification of a set of such known features appears to be a viable means of confirming a hypothetical registration transformation obtained from other evidence.

Some problems arose in verifying weak features -- features which

could be made out by a human using global tonal context and perhaps semantics. These were driveways and rough field boundaries. Sometimes there were competing edges nearby, such as caused by buildings along a road. As a result, the gradient values of feature points selected by a human were often as low or lower than either the noise threshold or the gradient value on a competing edge. Lowering the threshold cannot be done by itself because the large tolerance in predicted location would provide many possible spurious gradient peaks along the profiles. A probable remedy is to use 2-D trend (shape) from the stored feature in order to better select the "correct" below threshold peak.

This should be tried in future experiments.

The philosophy and technique of servoeing deserves further consideration. Servoeing was installed because accurate location predictions were impossible due to small errors in locating the image origin and approximation error in the transform  $T_\alpha^{-1}$ . The belief is that local features can be used to achieve better matching than is possible using only a global alignment. (This is comparable to getting accurate targeting by combining inertial guidance with image correlation in the target area.) However, servoeing can confuse real differences in the feature with differences due to locational error.

The current servoeing scheme records the differences  $(\Delta x_i, \Delta y_i)$  between the locations  $(x_{p,i}, y_{p,i})$  predicted by  $T_\alpha^{-1}$  applied to the stored feature track and the locations  $(x_{o,i}, y_{o,i})$  observed by doing the profile peak detection previously discussed. A window of up to 5 of the most recent differences is saved and used to adjust the next prediction as follows.

(1) The next prediction is made by transforming the next point on the stored curve and then adjusting it according to the window of differences.

$$(x_{p,n}, y_{p,n}) = T_{\alpha-1} (u_n, v_n) + 1/N \sum_{i=1}^N (\Delta x_i, \Delta y_i)$$

where  $N \leq 5$ .

(2) The next observed curve point  $(x_{o,n}, y_{o,n})$  is obtained by performing peak detection on the gradient profile perpendicular to the curve and through point  $(x_{p,n}, y_{p,n})$ .

(3) The difference between the prediction and observation is recorded as the most recent difference in the window (queue) and the least recent difference is purged.

$$(\Delta x_n, \Delta y_n) = (x_{o,n}, y_{o,n}) - (x_{p,n}, y_{p,n})$$

Step 1 weights all differences in the window equally, a policy which might be changed later. Step 2, however, presents the most opportunity for improvement. Perhaps the peak detection scheme should also consider the trend of the points in the window. In this manner the tracking should be less likely to go astray from the path with best global shape. This should be tried in future experiments.

In the future more attention will have to be paid to the interpretation of the differences between the predicted and observed curve points. Accuracy analysis must be done in order to get an error value to be used in interpreting the RMS value gotten from the verifier. Also, the patterns of missing points should be scrutinized to detect structured changes between the current imagery and the mapped imagery.

AD-A101 319

L N K CORP SILVER SPRING MD

F/G 9/2

KNOWLEDGE-BASED IMAGE ANALYSIS: (U)

APR 81 G C STOCKMAN, B A LAMBIRD, D LAVINE

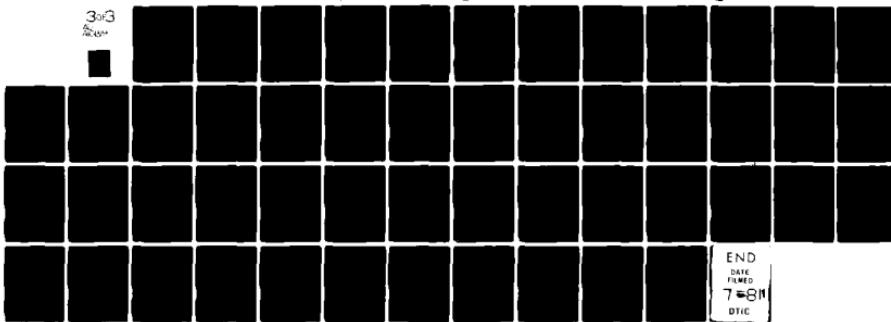
DAAK70-77-C-0110

UNCLASSIFIED

ETL-0258

NL

3x3  
25mm



END  
DATE FILMED  
7-6-81  
DTIC

## 6. Summary of Conclusions

LNK's registration procedure appears very promising. The procedure will provide the full RS&T transformation without an initial approximate transformation. The procedure seems to be adaptable to 3-D imagery which can be modelled. LNK's procedure can work with a wide range of features and is thus widely applicable. In particular, abstract vectors formed between point features show promise in being the most useful in rapidly providing an accurate transformation.

Since image to GDB registration did succeed in many cases with a very weak representation of the image, feature extraction can be less reliable than what is being attempted by most A.I. workers. Results reported here strongly support the potential for map-guided image analysis. Future testing is required to diversify the imagery handled, particularly to try images with little or no man-made structures. Accordingly other features will have to be detected and extending the Hough detector to handle more complex shapes is recommended. It is recommended that use of region features be added and pass points be identified along arbitrary boundary curves, such as high curvature points.

The intersection classification procedure worked very well on a limited set of non-aerial imagery. It is recommended the procedure be tested on a variety of imagery in order to determine its reliability. In addition the routines should be optimized to meet time and space requirements and should be extended to include intersections of curved lines.

Models for cartographic classes provide a means for disambiguating cartographic overlays. The disjoint principal components model was investigated as a means for class modelling. This method was selected both because it provides a natural measure of the strength of membership of a sample in each class modelled, and because it has proven useful in other fields such as chemistry.

Some simulation studies were performed on synthetic data to allow for comparison of the full ranking of class labels as provided by the disjoint principal components model, the pooled covariance model and the true density functions. This comparison would not have been possible with real data unless the underlying class distributions were known. While the disjoint principal components model performed poorly in the simulations, the results may not be indicative of their usefulness when using cartographic data. In fact, the classes were easily separated on the limited cartographic data available.

Therefore, we recommend extensive testing of this procedure on real data to provide a basis for evaluation. In addition, although the disjoint principal components model may not be suitable for cartographic domains, the idea of modeling each class separately should be explored further. In particular, methods such as clustering coupled with measures of the distance from a sample to a cluster could be used to model classes.

Various disambiguation procedures may be applied to provide a unique label for each region from a set of possible labels. A graph theoretical region matching procedure, the region adjacency graph, and a symbolic region matching method were investigated. Initial encouraging results on limited synthetic data indicate that these procedures should be explored further on more realistic problems.

Using servoeing techniques, the verification of lineals was successfully performed in a few images. However, much more diverse testing is recommended to assess the true potential of the technique. Boundaries of regions such as land/water and forest/field should be tried.

It is recommended that methods be devised for verifying region features in the image and that knowledge of elevation be included in order to augment pure grey-scale information. As much more needs to be learned about interpretation of the verification results with respect to change detection. It is recommended

that individual problems be studied and perhaps feature specific decision rules tried. Verification of image region labels is a complex problem involving the weighing of evidence from several sources such as texture measures, intersection type descriptions, region shape descriptions, location, and neighborhood context. We recommend a study of the feasibility of a knowledge based interactive system for verification. This study would entail detailed analysis of constraints on image regions and methods for structuring these constraints for inference.

## References

1. H.G. Barrow, J.M. Tenenbaum, R.C. Bolles and H.C. Wolf, Parametric Correspondence and Chamber Matching, Two new techniques for image matching, Proceedings of the 5th International Joint Conference on Artificial Intelligence, M.I.T., Aug. 22-25, 1977.
2. A. Bondy and L. Murty, Graph Theory with Applications, American Elsevier Publishing Co., New York, 1976.
3. M.A. Crombie, Semiautomatic Pass Point Determination Using Digital Techniques, U.S. Army Engineers Topographic Laboratories Report, ETL-0051, 1975.
4. W. Dixon and M. Brown, BMDP-79 Biomedical Computer Programs P-Series, University of California Press, Berkley, 1979.
5. R. Duda and P. Hart, Use of the Hough Transform to Detect Lines and Edges in Pictures, CAOM Vol. 15 No. 1 Jan. 1972, pp. 11-15.
6. L.R. Ford and D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, N.J., 1962.
7. B. Horne and B. Bachmann, Using Synthetic Images to Register Real Images with Surface Models, M.I.T., A.I. Memo 437, August, 1977.
8. T. Kanade, Region Segmentation: Signal vs. Segmentation, Computer Graphics and Image Processing, Vol 13, 1980, pp. 279-298.
9. A Kshirsagar, Multivariate Analysis, Marcel Dekker, Inc., New York, 1972.
10. D. Marr and E. Hildreth, Theory of Edge Detection, A.I.M. 518, M.I.T., April, 1979.
11. D. Milgram, Edge Point Linking Using Convergent Evidence, Proceedings of the Image Understanding Workshop, Arpa, November, 1978.
12. A. Nijenhuis and H. Wulf, Combinatorial Algorithms, Academic Press, New York, 1975.
13. T. Pavlidis, Structural Pattern Recognition, Springer Verlag, New York, 1977.
14. K. Price, Change Analysis and Scene Detection in Multispectral Images, Ph.D. Thesis Carnegie-Mellon University, Pittsburg, PA, Dec. 1976.
15. K. Price and R. Reddy, Matching Segments of Images, IEEE Vol. PAMI-1, pp. 110-116, 1979.
16. E.M. Riseman and M.A. Arbib, Computational Techniques in the Visual Segmentation of Static Scene, Computer Graphics and Image Processing Vol. 6, 1977, pp. 221-276.
17. G. Stockman and A. Agrawala, Equivalence of Hough Curve Detection and Template Matching, CAOM Vol. 20, No. 11, November, 1977.

18. G.C. Stockman and S.H. Kopstein, The Use of Models in Image Analysis, Report #AMRL-TR-78-117. Final Report by L.N.K. Corp. under Contract F33615-76-C-0521 to Aerospace Medical Research Lab., Wright-Patterson, A.F.B., Ohio 45433, Jan. 1979.
19. P. Van Wie and M. Stein, A Landsat Digital Image Rectification System, IEEE Trans. Geoscience Electronics, Vol. GE-15, No. 3, (July 1977).
20. S. Wold, Pattern Recognition By Means of Disjoint Principal Components Models, Pattern Recognition Vol. 8, 1976, pp. 127-139.
21. S. Zucker, Region Growing: Childhood and Adolescence, Computer Graphics and Image Processing, Vol. 5, 1976, pp. 382-399.
22. S. Zucker, R. Hummel, A. Rosenfeld, An Application of Relaxation Labeling to Line and Curve Enhancement, IEEE TC C-26, April, 1977.
23. S. Zucker and R. Hummal, Toward a Low-Level Description of Dot Clusters: Labeling, Edge, Interior and Noise Points, Computer Graphics and Image Processing, Vol. 9, No. 3, 1979, pp. 213-234.

## Appendix A Software Overview

The basic software used to support the research documented in this report was developed over a 3 year period on a Univac 1108 computer and then converted to run on a Hewlett-Packard minicomputer. There were substantial changes that had to be made due to word size differences, memory size differences, compiler differences, and the exciting change from access to pre-scanned digital pictures to on-line picture sampling. Nearly all driver programs were entirely rewritten and some enhancements were made to subroutines.

L.N.K. software currently in place at the E.T.L. ROSA Lab can be subdivided into 4 major subsystems which are detailed in appendices B,C,D, and E. These subsystems are as follows.

### A.1 Research Image Processing System (RIPS or RPS)

RIPS consists of routines which operate on grey scale imagery or gradient imagery. Their main objective is to extract edge features from gradient imagery. The most centrally used routine is the Hough straight edge detector. There are also routines for visual display and histogramming. RIPS data structures reside in a set of 6 named common blocks.

/RPSAC/ contains the accumulator array and auxiliary information for performing the Hough transform.

/RPSCC/ contains definitions of codes for linking a point with another point in its neighborhood thus forming generalized Freeman chains.

/RPSGC/ contains the gradient image and auxiliary information such as the gradient histogram and mask definitions.

/RPSIC/ contains the last sampled 32x32 image and auxiliary information.

/RPSSC/ contains RIPS system parameters and other globally used information such as intermediate I/O flags, current sampling window definition, and the DCB and buffer for I/O with the edge element file EEFILE.

/RPSWC/ contains working space for a list of selected image points for use by the feature detectors

These common blocks are added to program modules as macros which are given in appendix B to document in detail the common block content and the routines that operate on them.

#### A.2 Edge extraction routines EDGE<sub>X</sub> and EDGE<sub>Y</sub>

EDGE<sub>X</sub> and EDGE<sub>Y</sub> are two programs that extract lineal features from on-line imagery. EDGE<sub>X</sub> extracts straight and curved edge elements from single 32x32 windows. Window settings are gotten from a patterned scan of the image -- usually a raster scan of butting windows is used to cover the image but more flexible scanning is possible. The smart routines called are LHOUGH and LKTRKR which extract straight and curved edges respectively. Records of detections are written out to file EEFILE for plotting, registration, and/or input to EDGE<sub>Y</sub>.

EDGE<sub>Y</sub> takes the primitive detections made by EDGE<sub>X</sub> as input from file EEDGE<sub>X</sub> and attempts to extend straight edge elements and detect intersections along them. The gradient masking techniques are heavily used by EDGE<sub>Y</sub> to enhance detection of desired edges or to suppress competition from known edges. Records of extended edges or detected

intersections between edges are written out to file EEFILE. Both EDGEX and EDGEY require about 15 seconds of processing time per window sampled due to gradient computation time. More details appear in appendix C.

A.3 Registration software REG.

The registration software contains no image processing operators. Input to REG consists of image and map abstractions which are currently just a set of vectors. Output from REG consists of a listing of possible registration transformations (rotation and translation) along with an evaluation of their merit. At the heart of REG is a set of routines that perform a clustering operation in 3-space. The clustering is done by binning points into a set of 10x10x10 bins. 2,3, or 4 iterations of binning are typically required to get the bin size down to the size of error allowed in the final transformation parameters. Appendix D has details.

#### A.4 Verification software VERIFI

A fourth program and auxiliary routines exist for top-down verification of lineals in an image. Lineals are known a priori from a map or from previous image analysis and are to be referred to raw imagery via the registration transform for verification. The software transforms each point along the lineal using the registration transform and then performs a search for a match to the point by examining gradient profiles perpendicular to the direction of the curve. A match measure is built up for the entire lineal by averaging the squared distance between the predicted and observed locations of individual points. A special feature of VERIFI is its ability to make local adjustments (servo) to the predicted location of points based on past observations. This allows for legitimate departure from strictly linear registration transformations due to image deformations, relief displacement or actual movement of the feature over time. Details of the verification software are in Appendix E.

## Appendix B: RIPS Software

The Research Image Processing System (RIPS or RPS) is composed of routines that operate on digital images for the purpose of feature extraction or display. While there are driver programs which organize these routines for interactive exploration of imagery, the primary function of RPS software is as a subroutine library for image processing programs such as those documented in Appendices B,C, and E.

Routine	Arguments	Function
ARRAY array sampling	(IDELX, IDELY, IGXLO, IGYLO)	This routine moves the film stage to (IDELX, IDELY) relative to the current position and then obtains the 32x32 sample window. (IGXLO, IGYLO) is returned as the updated global stage position. ARRAY also updates the image common block with details about the newly taken sample.
CNVTPR polar to rectangular conversion	(THETA, R, NCOLS, NROWS, NPTS , X1, Y1, X2, Y2)	CNVTPR converts a Hough detection from polar form (THETA, R) to a directed edge from point (X1, Y1) to point (X2, Y2) which lie on the boundary of the sampling window. The window is defined as NROWSx NCOLS which is typically 32x32 for the ROSA lab. The darker side of the edge lies toward the right when traversing from (X1, Y1) to (X2, Y2). NPTS is returned as 2 iff conversion is successful.
FILTR edge masking	(AX, AY, BS, BY, CX, CY TYPE)	Used to filter or mask out an existing edge (AX, AY) to (BX, BY) from the current window. (CX, CY) is the current array center in global stage coordinates. TYPE specifies what kind of masking is to be done by the next gradient operators to be applied as follows.

Routine	Arguments	Function
		<p>TYPE=1 to mask out all points with same gradient direction as edge</p> <p>TYPE=2 to mask out all points with same direction or <math>180^\circ</math> + same direction as edge</p> <p>TYPE=3 all of the above + mask out all points along the edge (AX,AY) (BX,BY) regardless of gradient direction.</p> <p>Masks are actually set in gradient image common by calling routine MASKS.</p>
FIXANG	(RESLTN)	This routine sets up the accumulator array in preparation for performing the Hough transform. RESLTN= 8,16, or 32 depending on what angular resolution is being used for gradient direction calculations. The RIPS accumulator common block is initialized for use by routine LHOUGH.
GRADDL	(XLO,XHI,YLO,YHI, TYPE)	Computes a 32 directional gradient at each point within the subwindow X $\in$ (XLO,XHI) Y $\in$ (YLO,YHI). Iff TYPE=0 then thinning is also performed on the resulting gradient image. The resulting gradient image is stored in gradient image common.
HISTOG	(TYPE)	Computes histogram of image grey scales (if TYPE="IM") or gradient image (if TYPE="GR"). The histogram is stored in image common or gradient image common for use by SELECT or may be optionally printed.
LFOCUS	(NHITS,ITHRES,IRESL)	Used to focus Hough detections made with a coarse $11^\circ$ angular resolution to a $2^\circ$ angular resolution. The list of NHITS coarse detections is passed to LFOCUS from LSCAN through accumulator common. ITHRES is the threshold for the original detection. If the strength of the fine detection falls below 2/3 of ITHRES the detection is discarded. IRESL is the gradient direction resolution which is 8,16, or 32. Detections are written to file EEFILE.

Routine	Arguments	Function
LHOUGH Hough line detector	(XCENTR,YCENTR)	Computes the Hough transform for straight edge detection. Details of the detector are assumed to be set up in accumulator common by FIXANG. The input points are taken from a list in work common. Output is the accumulator array. (XCENTR,YCENTR) define the origin of the coordinate system to be used and is typically the window center (16,16). Detection of peaks in the accumulator array is left for LSCAN.
LKTRKR continuous curve detector	(XLO,XHI,YLO,YHI, LTHRSH)	Detects sets of high gradient points organized into continuous curves. Points have been linked by WRKLNK in the gradient image. (XLO,XHI,YLO,YHI) define the search window and LTHRSH defines the minimum number of points which make up a curve. Detected curves are chain-encoded and written out to file EEFILE.
LSCAN accumulator array peak detector	(NTHRES,NDET)	Detects local maxims in the accumulator array which exceed threshold NTHRES. Up to 5 detections are reported in accumulator common and the number of detections is returned in NDET.
MASKED point masking	(X,Y,D)	Logical function returns true iff point (X,Y) with gradient direction D is to be masked out of consideration due to location (X,Y) or gradient direction D. A previous call to FILTR or MASKS must have been issued to define the masking.
MASKS mask setting	(IOP,I,J,K,L)	Used to build masks for location or gradient direction masking as follows. IOP=1 clears all masking IOP=2 mask off (X,Y) if to the right of halfplane defined by (I,J)-(K,L) IOP=3 mask off direction D if D $\leq$ (I,J) IOP=4 print out current mask definitions.

Routine	Arguments	Function
MLTDIR multiple resolution gradient	(X,Y,RES,MAG, DIR,IERR)	Computes directional gradient at point (X,Y) using directional resolution of RES=8,16, or 32. Magnitude MAG and direction DIR of the gradient are returned. IERR#0 if error encountered. (X,Y) must be 2,4, or 5 pixels off window boundary for computation of the 8,16, or 32 directional gradients respectively.
PRINTA point accum- ulator array	(NUNIT)	Prints out the contents of the accumulator array on I/O unit NUNIT. A utility routine for human useable output only.
PRINTI print grey scales	(XLO,XHI,YLO,YHI, CHAR)	Utility routine to print out human useable grey scale plots of the image (CHAR="IM") or gradient image (CHAR="GR") of subwindow defined by X G (XLO,XHI), Y G (YLO,YHI).
SELECT select heavy gradient points	(XLO,SHI,YLO,YHI, THRSHL)	Routine for automatic selection of a threshold THRSHL such that a fraction FRACT of the gradient values exceed THRSHL. HISTOG is called to get a histogram and then THRSHL is set from the histogram and FRACT. Points that exceed the threshold are then placed in work common for the detection routines.  Points are selected from subwindow defined by X G (XLO,XHI), Y G (YLO,YHI)
WRKLNK link curve points to neighbors	(NCIRCL,SRCANG, DRTHRS)	Finds best connecting point for each point in work common in both the forward and backward direction. Connections are used later by LKTRKR to extract long continuous chains indicating curved edges. NCIRCL defined size of neighborhood searched for connecting point. Connecting point must lie within the angular range defined by SRCANG and have gradient direction within DRTHRS of the first point. Thus SRCANG and DRTHRS are used to control allowed curvature.

\*\*\*\*\*RIPS ACCUMULATOR ARRAY H.P. VERSION 29DEC78\*\*\*\*\*

C  
C INTEGER RLO,RHI,RJUMP,ACC,NR,NTHETA,TWTDIH  
C INTEGER DGRFLG,LOPTR,HIPT,FININO  
C REAL COSINE,SINE,KTHETA  
C INTEGER RADIUS,ANGLE,WEIGHT,WGDUMY  
C  
C COMMON AREA IS 1664 16-BIT WORDS LONG, OR 13 128 WORD BLOCKS  
C COMMON /RPSAC/ LENACC,WGDUMY,RLO,RHI,RJUMP,TWTDIH,NR,NTHETA,  
C + DGRFLG,LSCNUM,RADIUS(5),ANGLE(5),WEIGHT(5),LOPTR(32),  
C + HIPT(32),FININO(32),  
C + COSINE(64),SINE(64),KTHETA(64),ACC(32,33)  
C  
C ACC ACCUMULATOR ARRAY FOR THETA-RADIUS SPACE  
C KTHFTA VALUES FOR POSSIBLE ANGLES IN DEGREES  
C FININO FOR TRANSLATION BETWEEN ANGLE INDEX AND FINI DIRECTION  
C DGRFLG INDICATES IF ANGLES IN DEGREES ( ) OR FINIS ( )  
C THETA VALUE OF ANGLE SET IN REAL RADIAN  
C RLO,RHI,RJUMP DO-LOOP DEFINITION OF RADIUS VALUES  
C NR NUMBER OF RADIUS VALUES  
C NTHETA NUMBER OF ANGLES USED AND STORED IN KTHETA ARRAY  
C COSINE COSINES OF ANGLES IN KTHETA ARRAY  
C SINE SINES " " "  
C TWTDH WIDTH OF STRAIGHT LINE TEMPLATE  
C WGDHOU INDICATES WEIGHTED (=1) POINT OR WEIGHT 1 POINT (NOT 1)  
C LOPTR,HIPT POINTERS TO KTHETA ARRAY TO FILTER TRANSFORM ACC. TO  
C GRADIENT DIRECTION  
C RADIUS,ANGLE,WEIGHT PEAK RESPONSES FOUND IN ACCUMULATOR ARRAY  
C LSCNUM NUMBER OF PEAK RESPONSES "  
C  
C \*\*\*\*\*END ACCUM PROC H.P. VERSION 29DEC78 R.I.P.S.\*\*\*\*\*

C  
C IMAGE PROC H.P. VERSION 31 OCT 78 \*\*\*RIPS\*\*\*

C  
C STORAGE AND FORMAT FOR COMMON BLOCK CONTAINING IMAGE

C  
C INTEGER LENIMC,IMNAM,NCODES,NROWS,NCOLS,LEVELS,IMAG,NCOLWD  
C INTEGER HISTIM,IMGPAD,VXLO,VXHI,VYLO,VYHI,HISTFG  
C COMMON /RPSIC/ LENIMC,IMNAM(3),NROWS,NCOLS,NCODES,NCOLWD,  
C & LEVELS,VXLO,VXHI,VYLO,VYHI,HISTIM(64),  
C & IMGPAD,IMAG(32,32)  
C  
C LENIMC NUMBER OF WORDS IN COMMON (0 IF NOT INITIALIZED)  
C (ELSE LENIMC = 1100)  
C IMNAM ASCII NAME OF IMAGE  
C NROWS,NCOLS NUMBER OF ROWS AND COLUMNS OF IMAGE  
C LEVELS NUMBER OF GREY LEVELS USED (0,1,2,...,LEVELS-1)  
C IMAG IMAGE ARRAY  
C NCOLWD ARRAY MAPPING CONSTANT OF OLD UNIVAC RIPS  
C HISTFG HTSTOGRAM VALIDITY FLAG (NOT USED 30MAR79)  
C HISTIM HTSTOGRAM ARRAY  
C IMGPAD PADDING FOR PROPER EXEC CALL TO SAMPLE ARRAY  
C S... SCREEN WINDOW DEFINITION  
C V... VIRTUAL WINDOW DEFINITION (IN ABSOLUTE FILM COORDS)  
C  
C END IMAGE PROC \*\*\*RIPS\*\*\*

```

C CHSDFS  PROC    H.P. VERSION 22 NOV 78      ***RIPS***
C
C LTNE CHASTNG AND LTNKNG CODES FOR GENERAL FREEMAN CODE'S
C
C      INTEGER DX,DY,COMPMT
C      COMMON /RPSCC/ DX(68),DY(68),COMPMT(68)
C
C      DX      DX(LINK) IS X INCREMENT FOR FOLLOWING LTNK
C      DY      SIMILAR
C      COMPMT 180 DEGREE COMPLIMENT OF LINK
C
C CHSDFS  PROC    H.P. VERSION 22 NOV 78      ***RIPS***
FTN4,L,T
      BLOCK DATA RPSCH
C
C      INCLUDE RPSSYC
C      INCLUDE RPSIMC
C      INCLUDE RPSGRC
C      INCLUDE RPSWKC
C      INCLUDE RPSACC
C      INCLUDE RPSCHC
C
C      DATA DX /1,1,0,-1,-1,-1,0,1,
C      +      2,2,2,1,0,-1,-2,-2,-2,-1,0,1,
C      +      2,3,3,3,2,1,0,-1,-2,-3,-3,-3,-2,-1,0,1,
C      +      2,3,3,2,-2,-3,-3,-2,-1,0,1,4,4,4,1,0,
C      +      -1,-4,-4,-4,-4,-3,-2,2,3,4,4,3,2,-2,-3,-4/
C
C      59 45 46 47 60
C      58 44 34 35 36 37 61
C      57 43 33 18 19 20 21 38 62
C      56 32 17 6 7 8 9 22 48
C      55 31 16 5 * 1 10 23 49
C      54 30 15 4 3 2 11 24 50
C      68 42 29 14 13 12 25 39 63
C      67 41 28 27 26 40 64
C      66 53 52 51 65
C
C      DATA DY /0,-1,-1,-1,0,1,1,1,
C      + 1,0,-1,-2,-2,-2,-1,0,1,2,2,2,
C      + 2,1,0,-1,-2,-3,-3,-2,-1,0,1,2,3,3,3,
C      + 3,2,-2,-3,-3,-2,2,3,4,4,4,1,0,-1,-4,-4,
C      + -4,-1,0,1,2,3,4,4,3,2,-2,-3,-4,-4,-3,-2/
C      DATA COMPMT /5,6,7,8,1,2,3,4,15,16,17,18,19,20,9,10,11,12,13,14,
C      + 29,30,31,32,33,34,35,36,21,22,23,24,25,26,27,28,
C      + 41,42,43,44,37,38,39,40,51,52,53,54,55,56,45,46,
C      + 47,48,49,50,63,64,65,66,67,68,57,58,59,60,61,62/
C
C      COMPMT(0) IS UNDEFINED
C CHSDFS  PROC    H.P. VERSION 22NOV78      ***RIPS***
END

```

C GRADES PROC H.P. VERSION 31 OCT 78 \*\*\*RIPS\*\*\*

C FORMAT AND STORAGE OF COMMON STORING GRADIENT IMAGE

```
INTEGER LENGRC,GRNAM,HISTGD,HSXHI,HSXLO,HSYHI,HSYLO,HISTFL
INTEGER MAG,DIR,SUPRES,G(32,32),FLDIR(32,32),BLDIR(32,32)
INTEGER NTRPLS,NPAIRS,MKDIR1,MKDIR2
REAL MKLOCA,MKLOCB,MKLOCC
INTEGER RESLTN,WGTHOU
REAL HSMEAN,HSVAR
```

```
C COMMON /RPSGC/ LENGRC,GRNAM(3),RESLTN,WGTHOU,HSMEAN,HSVAR,
& HSXLO,HSXHI,HSYLO,HSYHI,HISTFL,HISTGD(64),
& MAXMAG,
& MAG(32,32),DIR(32,32),SUPRES(32,32),
& NTRPLS,MKLOCA(4),MKLOCB(4),MKLOCC(4),
& NPAIRS,MKDIR1(4),MKDIR2(4)
EQUIVALENCE (G,MAG),(FLDIR,DIR),(BLDIR,SUPRES)
```

C	LENGRC	NUMBER OF WORDS IN COMMON (0 IF NOT INITIALIZED) (ELSE LENGRC = 3152 )
C	GRNAM	NAME OF IMAGE FROM WHICH GRADIENT EXTRACTED
C	RESLTN	RESOLUTION OF GRADIENT DIRECTION (8,16,OR 32)
C	WGTHOU	PARAMETER FOR LHOUGH TRANSFORMATION
C	HSMEAN,HSVAR	MEAN AND VAR OF NOISE IN GRADIENT
C	HS...	WINDOW ON GRADIENT IMAGE USED FOR HISTOGRAM
C	HISTFL	HISTOGRAM VALIDITY FLAG (NOT USED 30MAR79)
C	HISTGD	HISTGDGRAM OF GRADIENT IMAGE
C	MAG	GRADIENT MAGNITUDE ARRAY
C	MAXMAG	MAXIMUM GRADIENT VALUE+1 = #GRADIENT LEVELS
C	DIR	GRADIENT DIRECTION
C	SUPRES	SUPPRESSION FLAG USED IN THINNING
C	NTRPLS	NUMBER OF HALFPLANE-DEFINING TRIPLES
C	MKLOCA/B/C	A,B,C COEFFICIENTS DEFINING HALFPLANE MASK
C	NPAIRS	NUMBER OF PAIRS DEFINING GRADIENT DIRECTION MASK
C	MKDIR1/2	DIRECTION RANGE DEFINING GRADIENT DIRECTION MASK
C	G	ALIAS FOR GRADIENT MAG
C	FLDIR,BLDIR	FORWARD AND BACKWARD EDGE DIRECTION
C	END GRADIT PROC	***RIPS***

```

C SYSTEM PROC H.P. VERSION 31OCT78 ***RIPS***
C MODES,PARAMETERS DEBUG FLAGS FOR RIPS
C
C      INTEGER INUNIT,OTUNIT,DRUNIT,TMUNIT
C      INTEGER IDCHEF,FENAME,EEBUFR
C      INTEGER RIPMOD,DEBUG,LENSYC,PWIDTH SYINAM,SYNCOL,SYNROW
C      INTEGER SXLO,SXHT,SYLO,SYHT,SYUXLO,SYUXHI,SYVYLO,SYVYHI
C      COMMON /RPSSC/ LENSYC,INUNIT,OTUNIT,DBUNIT,TMUNIT,
C      &          PWIDTH,RIPMOD,SYINAM(3),SYNROW,SYNCOL,
C      &          SXLO,SXHI,SYLO,SYHI,SYUXLO,SYUXHT,
C      &          SYVYLO,SYVYHI,DEBUG(20),FENAME(3),IDCBEE(144),
C      &          EEBUFR(66)
C
C      LENSYC      LENGTH IN WORDS OF COMMON (0 IF NOT INTIALIZED)
C      ( ELSE LENSYS = 187 )
C      SYINAM      NAME OF CURRENT IMAGE
C      SYNCOL      NUMBER OF COLUMNS OF CURRENT WINDOW
C      SYNROW      NUMBER OF ROWS OF CURRENT WINDOW
C      INUNIT      INPUT UNIT NUMBER, TYPICALLY FOR UNIT 5 INPUT
C      OTUNIT      OUTPUT UNIT, TYPICALLY PRINT UNIT 6
C      TMUNIT      INTERACTIVE TERMINAL UNIT NUMBER, I.E. 1 OR 4
C      RIPMOD      RIPS MODE, I.E. BATCH=0 OR INTERACTIVE=1
C      DEBUG       ARRAY OF DEBUG, OR INTERMEDIATE I/O FLAGS, ONE OR
C      MORE FOR EACH RIPS MODULE
C      S...        SCREEN WINDOW ON IMAGE IN LOCAL COORDINATES
C      SYV...      VIRTUAL WINDOW OF IMAGE IN ABSOLUTE FILM COORDINATES
C      PWIDTH      WIDTH OF PRINT LINE IN CHARACTERS, =72 OR 132
C      FENAME      NAME OF EDGE ELEMENT FILE, SHOULD BE "EEFILE"
C      IDCBEF     DATA CONTROL BLOCK FOR EFILE, OPENED AND CLOSED BY DRIVER
C      EEBUFR     BUFFER FOR OUTPUT OF DETECTED EDGE ELEMENTS TO EFILE
C
C      END SYSTEM PROC ***RIPS***
C
C WORKCM PROC H.P. VERSION 31OCT78 ***RIPS***
C
C DEFINES WORK COMMON BLOCK WHERE SETS OF SELECTED POINTS STORED
C
C      INTEGER LENWKC,LENWRK,WX(128),WY(128),WGRDIR(128)
C      INTEGER WGR(128),WFLEDIR(128),WKNAME(3)
C      EQUIVALENCE (WFLEDIR,WGRDIR)
C      COMMON /RPSWC/ LENWKC,LENWRK,WKNAME,WX,WY,WGR,WGRDIR
C
C      LENWKC      TOTAL LENGTH IN WORDS OF COMMON (0 IF NOT INTIALIZED)
C      ( ELSE LENWKC = 517 )
C      WKNAME      ASCII NAME OF PICTURE FROM WHICH POINTS SELECTED
C      LENWRK      NUMBER OF POINTS SELECTED
C      WX,WY      X,Y COORDINATES OF SELECTED POINTS
C      WFLEDIR,WFLEDIR FORWARD AND BACKWARD EDGE DIRECTION AT POINT
C      WGRDIR      GRADIENT DIRECTION AT POINT
C      WGR        GRADIENT MAGNITUDE AT POINT
C
C      END WORKCM PROC ***RIPS***

```

Appendix C: Edge extraction routines EDGEY and EDGEY.

Two batch programs exist which extract lineal edge data from on-line imagery. EDGEY performs a patterned scan of the image extracting straight and curved edge elements from sampled windows. One record is written to file EEFILE for each edge element extracted. The scan pattern is defined by giving a window spacing in stagels, the beginning window center in stagels, and the trajectory of subsequent window centers in terms of a Freeman chain code. The following output gives a sample run of EDGEY scanning around a road intersection. Typically EDGEY is used by repeating a raster scan pattern across the entire image. EDGEY allows checkpointing after each patterned scan is complete, or the user can repeat the scan pattern over the next region of the image.

EDGEY requires about 15 seconds per sampled window; almost all of that time being consumed by the gradient operator. LHOUGH and LKTRKR are the smart routines called and extract straight or continuously curved edge elements respectively. Detections are written to EEFILE which must be saved by copying if the checkpointing facility is being used. The following output gives routines loaded with EDGEY and sample output of the program.

EDGEY is a batch program which takes the primitive detections made by EDGEY as input and attempts to extend straight edge elements and detect intersections along them.

For each primitive straight edge element input to EDGEY, the processing logic is as follows.

- 1) The stage is positioned so that the edge element should be centered in the window. Competing edge directions are masked off and the Hough detector is called to verify the edge. If the edge is missed (happens infrequently) then go to step 5. If the edge is detected adjust (servo) the stage to center the edge element in the window.
- 2) Search along the detected edge for nearly perpendicular intersecting straight edges. When searching, the existing edge direction  $d$  and  $d + 180^\circ$  are masked off to remove contention. Also the stage is toggled so that almost an entire window is visible on either side of the existing edge. (See Figure 2.10).
- 3) Attempt to extend the current edge in the forward direction. If a forward extension exists go to step 2, else continue at step 4.
- 4) Attempt to extend the current edge in the backward direction. If a backward extension exists perform step 2 search for intersecting edges and repeat this step. When no further backward extension is detected continue at step 5.
- 5) Negate any stage adjustment made in step 1. Write a record out to EEFILE defining the full extent of the edge found. (Any detected intersections were written out to EEFILE when detected.)

The chief defect in EDGEY as currently implemented is that the routine does not remember the edges it has already worked on. Quite often, EDGEX will detect several pieces of a long straight edge. EDGEY is then condemned to extend and search from each piece, often achieving multiple identical results. Theoretically the duplication is easy to remove at a later stage, but a great deal of time is currently wasted working over the same edge.

C.1 Loader commands for creating EDGEEX

Command	Comment
RE,%EDGEEX	main routine
RE,%ZARRAY	window sampling routine
RE,%RPSFX	routine initializes accumulator array
RE,%RPSGL	computes 32-directional gradient
RE,%RPSHS	histograms gradient
RE,%RPSSE	selects strong gradient points
RE,%RPSHG	computes Hough transform
RE,%RPSFO	focuses Hough transform
RE,%RPSLS	scans accumulators for peaks
RE,%RPSWR	links edge points to best neighbors
RE,%RPSLK	extracts continuous curves
RE,%RPSPR	prints grey scales if required
RE,%RPSPA	prints acc. array if required
RE,%RPSML	computes gradient at single point
RE,%RPSVG	converts polar vector to rectangular
RE,%STAGE	positions stage
RE,%RPSMK	sets gradient masks
RE,%RPSCH	block data routine for chain code common
RE,%RPSVT	converts vector to gradient direction
RE,%RPSAA	worker subroutine for RPSLS

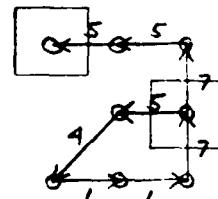
C.2 SAMPLE RUN OF EDGEX

```

:RUN,EDGEX,6,6
GIVE IMAGE NAME(3A2)-
4621-3
GIVE FRACTION OF POINTS TO SELECT(REAL/FREE)-
0.05,
GIVE 10 I/O FLAGS (INT/FREE)-
0,0,0,0,0,0,0,0,0,0,
GIVE GLOBAL COORDINATES OF ORIGIN (INT/FREE)-
5000,5000, must have stage positioned
# PTS ON LINE & # PTS ON CHAIN ? (2 INT/FREE)-
20,20,
FIRST WINDOW OFF ORIGIN-DELX,DELY (INT/FREE)-
6120,3072,
GIVE SAMPLING TRAJECTORY : LSIZE,N
(LINKS(I),I=1,N) (INT/60I1)
200,8,
54112755
IERR= 4 IN OPENING FILE EEFILE
IERR= 0 IN WRITING LABEL ON EEFILE
GRADDL ON IMAGE=4621-3 WINDOW=1100011248 7952 8200
GRADDL:NOW THIN MAXMAG,TYPE= 155 0
SELECT:THRESHOLD= 82
SELECT:LENWRK= 66
0 SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 20 WERE FOUND AT
ANGLE OR DIRECTION  RADIUS  WEIGHT
 17      -3      25
LFOCUS:11159, 8207 11159, 7959 179.0, -3.0, WT= 15 20,32 20, 1
LKTRKR: 0 CHAINS OUTPUT OF MIN LENGTH: 20 - 0 LINKS TOTAL
***VIRTUAL*** 11000 11248 7952 8200*SCREEN* 1 32 1 32 *****
1234567890123456789012345678901234567890
32
31
30
29
28
27
X

```

9 samples indicated  
by scan pattern.  
1st = 6th and 9th  
windows shown.



```

" (LINKS(I),I=1,N) (INT/60I1)
" 200,8,
54117755
" IERR= 4 IN OPENING FILE EEFILE
" IERR= 0 IN WRITING LABEL ON EEFILE
" GRADDL ON IMAGE=4621-3 WINDOW=1100011248 7952 8200
" GRADDL:NOW THIN MAXMAG,TYPE= 155 0
" SELECT:THRESHOLD= 82
" SELECT:LENWRK= 66
0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 20 WERE FOUND AT
" ANGLE OR DIRECTION RADIUS WEIGHT
" 17 -3 25
" LFOCUS:11159, 8207 11159, 7959 179.0, -3.0, WT= 15 20,32 20, 1
" LKTRKR: 0 CHAINS OUTPUT OF MIN LENGTH 20 - 0 LINKS TOTAL
" ***VIRTUAL*** 11000 11248 7952 8200*SCREEN* 1 32 1 32 *****
" 1234567890123456789012345678901234567890
" 32
" 31
" 30
" 29
" 28
" 27 X
" 26 XX
" 25 XX
" 24 X $X
" 23 X $X
" 22 X $X
" 21 XX X $X
" 20 $ $X
" 19 X $X
" 18 $ $X
" 17 $ $X
" 16 $ $XX
" 15 * $$$$X
" 14
" 13
" 12 $X
" 11 *
" 10 $
" 9 $ $X
" 8 $
" 7 X $
" 6 X X
" 5 X X
" 4
" 3
" 2
" 1
" GRADDL ON IMAGE=4621-3 WINDOW=1080011048 7952 8200
" GRADDL:NOW THIN MAXMAG,TYPE= 107 0
" SELECT:THRESHOLD= 50
" SELECT:LENWRK= 30
0SCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 20
" LKTRKR: 0 CHAINS OUTPUT OF MIN LENGTH 20 - 0 LINKS TOTAL
" GRADDL ON IMAGE=4621-3 WINDOW=1060010848 7752 8000
" GRADDL:NOW THIN MAXMAG,TYPE= 28 0
" SELECT:THRESHOLD= 18
" SELECT:LENWRK= 34
0SCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 20
" LKTRKR: 0 CHAINS OUTPUT OF MIN LENGTH 20 - 0 LINKS TOTAL
" GRADDL ON IMAGE=4621-3 WINDOW=1080011048 7752 8000
" GRADDL:NOW THIN MAXMAG,TYPE= 85 0
" SELECT:THRESHOLD= 11
" SELECT:LENWRK= 88
0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 20 WERE FOUND AT
" ANGLE OR DIRECTION RADIUS WEIGHT
" 23 2 22
" LKTRKR: 0 CHAINS OUTPUT OF MIN LENGTH 20 - 0 LINKS TOTAL
" GRADDL ON IMAGE=4621-3 WINDOW=1100011248 7752 8000
" GRADDL:NOW THIN MAXMAG,TYPE= 141 0
" SELECT:THRESHOLD= 74
" SELECT:LENWRK= 62
0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 20 WERE FOUND AT
" ANGLE OR DIRECTION RADIUS WEIGHT
" 17 -2 27
" LFOCUS:11159, 8007 11159, 7759 179.0, -3.0, WT= 22 20,32 20, 1
" LKTRKR: 0 CHAINS OUTPUT OF MIN LENGTH 20 - 0 LINKS TOTAL
" ***VIRTUAL*** 11000 11248 7752 8000*SCREEN* 1 32 1 32 *****
" 1234567890123456789012345678901234567890

```

0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 20 WERE FOUND AT

ANGLE OR DIRECTION	RADIUS	WEIGHT
17	-2	27
LFOCUS:11159, 8007	11159, 7759	179.0, -3.0, WT= 22 20,32 20, 1
LKTRKR:	0	CHAINS OUTPUT OF MIN LENGTH 20 - 0 LINKS TOTAL
***VIRTUAL***	11000 11248 7752 8000*SCREEN*	1 32 1 32 *****
1234567890123456789012345678901234567890		
32		
31		
30		
29		
28		
27		
26		X
25		\$
24		\$
23		\$
22		\$ \$X
21		\$ \$X
20		\$ \$X
19		\$ \$
18		\$
17		\$ \$
16		\$ \$
15		\$ \$
14	\$X	\$ \$
13	XX	\$ \$
12	\$X	\$ \$
11	##	\$ \$
10	##	X \$
9	##X	\$ \$
8	X	\$ \$
7		\$ \$
6		\$ \$
5	SSX	\$ \$
4		
3		
2		
1		

GRADDL ON IMAGE=4621-3 WINDOW=1100011248 7952 8200

GRADDL:NOW THIN MAXMAG,TYPE= 157 0

SELECT:THRESHOLD= 80

SELECT:LENWRK= 59

0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 20 WERE FOUND AT

ANGLE OR DIRECTION	RADIUS	WEIGHT
17	-3	24
LFOCUS:11159, 8207	11159, 7959	179.0, -3.0, WT= 14 20,32 20, 1
LKTRKR:	0	CHAINS OUTPUT OF MIN LENGTH 20 - 0 LINKS TOTAL
***VIRTUAL***	11000 11248 7952 8200*SCREEN*	1 32 1 32 *****
1234567890123456789012345678901234567890		
32		
31		
30		
29		
28		X
27		
26		XX
25		XX
24		X
23		X \$X
22		X \$S
21		X \$X
20		\$ \$S
19		X \$X
18		\$X
17		\$X
16		SSX
15		SSSX
14		
13		
12		
11		XX
10		\$ \$
9		\$ X
8		\$ \$
7		X \$
6		X X
5		X
4		

```

3
2
1
GRADDL ON IMAGE=4621-3 WINDOW=1100011248 8152 8400
GRADDL:NOW THIN MAXMAG,TYPE= 129 0
SELECT:THRESHOLD= 64
SELECT:LENWRK= 69
0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 20 WERE FOUND AT
ANGLE OR DIRECTION  RADIUS  WEIGHT
17      -4      45
1       2      24
LFOCUS:11167, 8407  11167, 8159      179.0,   -4.0, WT= 24  21,32  21, 1
LFOCUS:11143, 8159  11143, 8407      -1.0,    1.0, WT= 24  18, 1  18,32
LKTRKR:LINID,AX,AY,GLOBAL AX,AY=      1      17      5 11143  8199  23
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
7 7 7
LKTRKR: 1 CHAINS OUTPUT OF MIN LENGTH 20 - 0 LINKS TOTAL
***VIRTUAL*** 11000 11248 8152 8400*SCREEN* 1 32 1 32 *****
1234567890123456789012345678901234567890
32
31
30
29
28      X  X
27      X  X
26      $  $
25      $  $+
24      $  $+
23      $  $X
22      $  $X
21      $  $X
20      $  $X
19      $  $X
18      $  $X
17      $  $X
16      $  $X
15      $  $X
14      $  $X
13      $  $X
12      $  $X
11      $  $X
10      $  $X
9       $  $X
8       $  $X
7       $  $X
6       $  $X
5       $  $X
4
3
2
1
GRADDL ON IMAGE=4621-3 WINDOW=1080011048 8152 8400
GRADDL:NOW THIN MAXMAG,TYPE= 114 0
SELECT:THRESHOLD= 10
SELECT:LENWRK= 80
0SCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 20
LKTRKR: 1 CHAINS OUTPUT OF MIN LENGTH 20 - 0 LINKS TOTAL
GRADDL ON IMAGE=4621-3 WINDOW=1060010848 8152 8400
GRADDL:NOW THIN MAXMAG,TYPE= 156 0
SELECT:THRESHOLD= 73
SELECT:LENWRK= 35
0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 20 WERE FOUND AT
ANGLE OR DIRECTION  RADIUS  WEIGHT
11      4      24

```

```

* GRADDL: NOW THIN MAXMAG, TYPE= 156      0
* SELECT: THRESHOLD= 73
* SELECT: LENWRK= 35
* 0 SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 20 WERE FOUND AT
* ANGLE OR DIRECTION    RADIUS    WEIGHT
*      11        4        24
* LFOCUS: 10607, 8303  10855, 8199    247.0,    4.0, WT= 16  1,19  32, 6
* LKTRKR: LINID,AX,AY,GLOBAL AX,AY=      2        5        17  10647  8295  23
*      1  2  1  2  1  2  1  2  1  2  1  2  1  1  1  2  2  1
*      1  2  1
* LKTRKR: 2 CHAINS OUTPUT OF MIN LENGTH 20 -  0 LINKS TOTAL
* ***VIRTUAL*** 10600 10848 8152 8400*SCREEN* 1 32  1 32 *****
* 1234567890123456789012345678901234567890
* 32
* 31
* 30
* 29
* 28
* 27
* 26
* 25
* 24
* 23
* 22
* 21
* 20
* 19
* 18
* 17
* 16
* 15
* 14
* 13
* 12
* 11
* 10
* 9
* 8
* 7
* 6
* 5
* 4
* 3
* 2
* 1
* PATTERN COMPLETED  9 WINDOWS SAMPLED
* WANT TO CONTINUE SCAN PATTERN? (YES=1/NO=0)-
* 0,
* IERR=  0 IN CLOSING EEFILE

```

C.3 Loader commands for creating EDGEY

Command	Comment
RE,%EDGEY	main routine
RE,%RPSCH	block data routine for chain code common
RE,%ARRAY	window sampling routine
RE,%RPSFX	routine initializes accumulator array
RE,%RPSGL	computes 32-direction gradient
RE,%RPSHG	computes Hough transform
RE,%RPSFO	focuses Hough transform
RE,%RPSLK	extracts continuous curves
RE,%RPSLS	scans accumulators for peaks
RE,%RPSPR	prints grey scales if required
RE,%RPSPA	prints accumulators if required
RE,%RPSSE	selects strong gradient points
RE,%RPSWR	links edge points to best neighbors
RE,%RPSVT	converts vector to gradient direction
RE,%RPSHS	histograms gradient
RE,%RPSML	computes gradient at single point
RE,%RPSVG	converts polar vector to rectangular
RE,%STAGE	positions stage
RE,%RPSFR	sets up masks for given edge
RE,%RPSMK	maintains and applies gradient masks
RE,%INSPT	inspects both sides of edge for other edges
RE,%VEREE	verifies previously detected edge
RE,%INSEC	computes intersection between 2 lines
RE,%RPSAA	worker subroutine for RPSLS

C.4 SAMPLE RUN OF EDGEY

```

AT EEFIL1 : 10 EEDR01 : 10
:DU: EEDR01 : 10,11 EEDR1 : 10
:RUN,EDT06
SOURCE FILE?
/EDR01:10
  DREUX ORG=10000 5000PERCNT= .05WGT= 20 15
  +
  LK 8111, 2733 16, 7, 7, 6, 6, 7, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7,
  /-90
  FO 6943, 4885, 7023, 4637,
  /+5
  FO 9637, 5209, 9885, 5281,
  +
  LK 9677, 5225, 23, 1, 8, 1, 1, 8, 1, 1, 1, 8, 1, 1, 1, 8, 1, 1, 1, 1, 2
  /-900
EDF
/1
  DREUX ORG=10000 5000PERCNT= .05WGT= 20 15
/L50
  DREUX ORG=10000 5000PERCNT= .05WGT= 20 15
  FO 6943, 4885, 7023, 4637,
  FO 2281, 5135, 5281, 4887,
  FO 3885, 5103, 3637, 4959,
  LK10677, 4927, 17,20,20, 8, 8,20, 8,20,20, 8, 8, 8,20, 8,20, 8, 8,10,
  LK10677, 5201, 21, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 23, 6 20, 6
  FO 9637, 5209, 9885, 5281,
EOF
/ER
END OF EDIT
:RUN,EDGEY,6,6
GIVE IMAGE NAME(3A2)-
DREUX
  GIVE FRACTION OF POINTS TO SELECT(REAL/FREE)-
0.05,
  GIVE 10 I/O FLAGS (INT/FREE)-
0,0,0,0,0,0,0,0,0,0,
  GIVE GLOBAL ORIGIN AND STAGE BOUNDS (6 INT/FREE)-
10000,5000,850,2700,11200,9200,
  # PTS ON LINE, ON CHAIN, & TRACKING SIZE (3 INT/FREE)=
15,15,150,
IERR= 4 IN OPENING FILE EEFIL1
IERR= 4 IN OPENING FILE EEDGEY
IERR,LEN= 0 23
EEFILE REC=
DREUX ORG=10000 5000PERCNT= .05WGT= 20 15
IERR,LEN= 0 13
EEFILE REC=
FO 6943, 4885, 7023, 4637,
GRADDL: ON DREUX WINDOW= 6863 7111 4641 4889 MAXMAG,TYPE= 62 0
SELECT:THRSHL,LENWRK= 29 31
0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 15 WERE FOUND AT

  ANGLE OR DIRECTION  RADIUS  WEIGHT
    15      2      31
LFOCUS: 6934, 4896 7030, 4648 200 0, 2.0, WT= 20 9.32 21, 1
GRADDL: ON DREUX WINDOW= 6934 7182 4693 4941 MAXMAG,TYPE= 21 0
SELECT:THRSHL,LENWRK= 8 11
0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15
GRADDL: ON DREUX WINDOW= 6750 6998 4693 4941 MAXMAG,TYPE= 31 0
SELECT:THRSHL,LENWRK= 8 17
0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15
GRADDL: ON DREUX WINDOW= 6980 7228 4550 4798 MAXMAG,TYPE= 22 0
SELECT:THRSHL,LENWRK= 8 9
0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15
GRADDL: ON DREUX WINDOW= 6796 7044 4550 4798 MAXMAG,TYPE= 30 0
SELECT:THRSHL,LENWRK= 9 34
0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15
GRADDL: ON DREUX WINDOW= 6943 7191 4393 4641 MAXMAG,TYPE= 45 0
SELECT:THRSHL,LENWRK= 25 32
0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 15 WERE FOUND AT

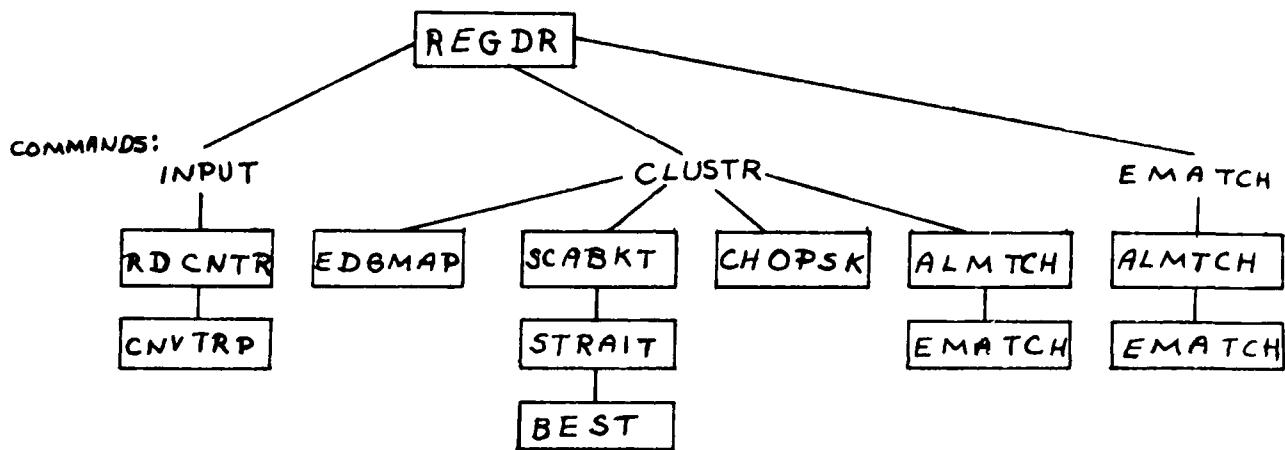
  ANGLE OR DIRECTION  RADIUS  WEIGHT
    15      -1      32
LFOCUS: 7038, 4648 7142, 4400 202.0, -1.0, WT= 22 12.32 25, 1
FORWARD VEREE:TAX,TAY,TBX,TBY,KFLAG= 7023 4637 7103 4389 1
GRADDL: ON DREUX WINDOW= 7014 7262 4445 4693 MAXMAG,TYPE= 17 0
SELECT:THRSHL,LENWRK= 8 12
0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15
GRADDL: ON DREUX WINDOW= 6830 7078 4445 4693 MAXMAG,TYPE= 26 0
SELECT:THRSHL,LENWRK= 8 22
0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15
GRADDL: ON DREUX WINDOW= 7060 7308 4302 4550 MAXMAG,TYPE= 23 0
SELECT:THRSHL,LENWRK= 8 13

```

0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15  
 GRADDL: ON DREUX WINDOW= 6976 7134 4102 4160 MAXMAG,TYPE= 14 0  
 SELECT:THRSHL,LENWRK= 11 1  
 0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15  
 GRADDL: ON DREUX WINDOW= 7023 7271 4145 4193 MAXMAG,TYPE= 14 0  
 SELECT:THRSHL,LENWRK= 29 32  
 0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 15 WERE FOUND AT  
 ANGLE OR DIRECTION RADIUS WEIGHT  
 15 -2 32  
 LFOCUS: 7134, 4400 7222, 4152 200.0, -2.0, WT= 21 14.32 25. 1  
 FORWARD VEREE:TAX,TAY,TBX,TBY,KFLAG= 7103 4389 7183 4141 1  
 GRADDL: ON DREUX WINDOW= 7094 7342 4197 4445 MAXMAG,TYPE= 28 0  
 SELECT:THRSHL,LENWRK= 8 15  
 0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15  
 GRADDL: ON DREUX WINDOW= 6910 7158 4197 4445 MAXMAG,TYPE= 36 0  
 SELECT:THRSHL,LENWRK= 8 30  
 0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15  
 GRADDL: ON DREUX WINDOW= 7140 7388 4054 4302 MAXMAG,TYPE= 30 0  
 SELECT:THRSHL,LENWRK= 8 25  
 0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15  
 GRADDL: ON DREUX WINDOW= 6956 7204 4054 4302 MAXMAG,TYPE= 28 0  
 SELECT:THRSHL,LENWRK= 8 29  
 0LSCAN: NO LOCAL MAXIMA FOUND WITH THRESH = 15  
 GRADDL: ON DREUX WINDOW= 7103 7351 3897 4145 MAXMAG,TYPE= 24 0  
 SELECT:THRSHL,LENWRK= 15 42  
 0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 15 WERE FOUND AT  
 ANGLE OR DIRECTION RADIUS WEIGHT  
 15 -4 37  
 LFOCUS: 7222, 4152 7334, 3904 204.0, -4.0, WT= 18 15.32 29. 1  
 FORWARD VEREE:TAX,TAY,TBX,TBY,KFLAG= 7183 4141 7263 3893 1  
 GRADDL: ON DREUX WINDOW= 7174 7422 3949 4197 MAXMAG,TYPE= 28 0  
 SELECT:THRSHL,LENWRK= 15 59  
 0SIGNIFICANT LOCAL MAXIMA WITH THRESHOLD = 15 WERE FOUND AT  
 ANGLE OR DIRECTION RADIUS WEIGHT  
 20 -9 27  
 LFOCUS: 7429, 4124 7301, 3956 143.0, -9.0, WT= 14 32.22 16. 1  
 \*\*\*VIRTUAL\*\*\* 7174 7422 3949 4197\*SCREEN\* 1 32 1 32 \*\*\*\*\*  
 1234567890123456789012345678901234567890  
 32  
 31  
 30  
 29  
 28  
 27  
 26  
 25  
 24  
 23  
 22  
 -----  
 21 // -  
 20 -- /  
 19 -  
 18 /  
 17 + -  
 16 x + /  
 15 x + /  
 14 \$ \$x+-  
 13 \$ x  
 12 \$ +  
 11 \$ \$x/  
 10 \$ x/-  
 9 - x/  
 8 - x +  
 7 - x + -  
 6 x /  
 5 / / + / / \$  
 4  
 3  
 2  
 1  
 INSEC RETURNS AX,AY,BX,BY,CX,CY,DX,DY,IX,IY,ITYPE= 7183 4141 7229 3998 7429 4140 7293 3956 7258 3908

Appendix D Registration Software

The registration software consists of one load module containing one main program and several subroutines. The program structure is illustrated below:



The registration program takes a set of image edges from the file IMAGE and a set of map edges from the file MAP and computes the possible transformations that will transform the image onto the map. The technique used is to cluster the possible transformations in 3-dimensional space and select those with the strongest support.

The data structures used reside in the four common blocks, presented in D.1 , and the match weight matrix, MATCHS(200,30) is in the common block MATRX which resides in the RTE-IV, extended memory facility.

Routine Name	Arguments	Function
REGDR	Main Program	Reads and analyzes commands and calls appropriate subroutine. Sets up initial values for variables contained in common blocks found in the procs: REGIST, BUCKET, and PRTFLG.
RDCNTR	(NUMLEV,SCALEX, SCALEY,SCALET,LOXBND, LOYBND,LOTBND)	Reads in the edge information for the image and then the map from DISK FILES. Image edge information must reside in the file IMAGE::18. Map edge information must reside in MAP::18. Computes centers of image and map windows and places them in common block. Using NUMLEV, computes the three scales at each level and stores them in arrays SCALEX,SCALEY and SCALET. Computes lower bounds at first level and stores them in LOXBND(1), LOYBND(1) and LOTBND(1). Image edge information stored in arrays X1,X2,Y1,Y2 of common block REGIS. Map edge information into arrays U1,U2,V1,V2 of same common block.
CNVTRP	(X1,Y1,X2,Y2,RADIUS, THETA)	Takes a directed straight line segment beginning at (X1,Y1) and ending at (X2,Y2) and converts it to polar coordinates (RADIUS,THETA).
EDGMAP	(X1,Y1,X2,Y2,U1,V1, U2,V2,STOL,THETAR, STAILX,STAILY,SHEADX, SHEADY)	Given an image edge defined by (X1,Y1) and (X2,Y2), a map edge defined by (U1,V1) and (U2,V2), an angle of rotation of image to map edge (THETAR) and a tolerance (STOL); a line segment in $\alpha$ -space which represents the constraints of the x-shift and the y-shift in transformations on these two edges is calculated and is represented by (STAILX, STAILY) and (SHEADX,SHEADY).
SCABKT	(TAILX,TAILY,HEADX, HEADY,THETAI,THEIND, SCALX,LOX,SCALY,LOY, MATNUM,OUTUNT)	Creates the clustering matrices used to find the most relevant transformations. Since two sets of clustering matrices are used by the approach (regular and offset matrices) the routine is called twice for each line segment it processes. The line segment is defined by (TAILX, TAILY),(HEADX,HEADY) and the angle of

Routine Name	Arguments	Function
		the line is in THETAI. MATNUM indicates whether incrementation is to be done in the regular matrix (MATNUM=1) or the offset matrix (MATNUM=2). The scale of the matrix is defined by SCALX, LOX, SCALY and LOY. The line is chased from tail to head and each bucket of the matrix that it passes through is incremented. Information about each line is written to an intermediate output file. Clustering matrices BKTcnt & BKTOFF are in the common block in proc BUCKET.
STRAIT	(AX,AY,BX,BY, NNEIGB,STORAG,N,NLINKS, IFLAG)	Given a line defined by (AX,AY) and (BX,BY) and whether it is to use 8-directional or 4-direction Freeman codes. STRAIT computes the links along that straight line. The links are stored in the array STORAG(N). If the number of links, NLINK, is greater than N, IFLAG is set to indicate an error.
BEST	(FX,FY,TX,TY,NNEIGE L1,L2,N1,N2)	Given a line from (FX,FY) to (TX,TY); computes the straightest path, returning # of L1 links in N1 and # of L2 links in N2.
CHOSPK	(PEAKS, MAXPKS,NUMPKS, PRTFLG,PWIDTH,THRESH)	Scans the two clustering matrices, BKTcnt and BKTOFF, to find the MAXPKS highest values. The indices of the high valued buckets are stored in the array PEAKS with the number of peaks actually chosen being in NUMPKS. THRESH defines the minimum value to be considered as a possible peak. PRTFLG and PWIDTH provide information to the print routines.
ALMTCH	(THETA,XSHIFT,YSHIFT, ANGTOL,DTOL,PWIDTH, LENCHK,MTCHWT, NMCHRW,NMCHCL,COLMCH)	Given a transformation defined by (THETA,XSHIFT,YSHIFT), an angle tolerance of ANGTOL and a distance tolerance of DTOL; an evaluation of how good the transformation is made. This is done by transforming any image edge that possess approximately the correct gradient onto a map edge. The function EMATCH is

Routine Name	Arguments	Function
		called to actually evaluate how good each match is. Taking the results of EMATCH, ALMTCH creates the match weight matrix matchs (in common block MATRX), the average match weight, MTCHWT, the number of matching rows, NMCHRW, the number of matching columns, NMCHCL, and in which columns a match has been found. If LENCHK=1, only edges of approximately the same length are compared. PWIDTH is used by available output routine.
EMATCH	(SEG1,SEG2,GRAD,THETA, XSHIFT,YSHIFT,DTOL)	Given the transformation (THETA,XSHIFT, YSHIFT), the edge in the image (stored in array SEG1), the edge in the map (stored in array SEG2), the gradient direction of the map edge, GRAD, the gradient direction of the image edge, THETA, and the tolerance, DTOL; computes strength of match and returns a value between 0 and 1.
Worker routines not shown in program structure.		
STACK	(CODE,PEAKS,I)	Maintains a push down stack that keeps track of peaks to be evaluated. Each entry contains 5 fields; THETA INDEX, X-INDEX, Y-INDEX, WEIGHT and which matrix peak appeared in.
PRTBKT	(PWIDTH,CNTSUM)	Prints either the original or offset clustering matrix depending on the value of CNTSUM. Matrices in common block BUCKET
PRTMAT	(RSIZE,CSIZE,NIMAGE, NMAP,PWIDTH)	Prints the match weight matrix created in ALMTCH.
RDWRTF	(FILENO,OPCODE, BUFFER,IL,IERR, TYPE)	Controls the input and output to the intermediate disk files. Possible operations are OPEN, READ, WRITE,REWIND, and CLOSE. Can have up to 2 files open at once.
SGN	(I)	Returns 1 if I > 0, -1 if I < 0 and 0 if I=0.

D.1 Procs containing common blocks used in registration

```
C
C REGIST PROC*****H.P. REGISTRATION CODE 9FEB79*****
C
C     INTEGER X1(200),X2(200),Y1(200),Y2(200),U1(30),
C             * U2(30),V1(30),V2(30),NIMAGE,NMAP,IXCENT,IYCENT,MXCENT,
C             * MYCENT,RSIZE,CSIZE
C             REAL THET(2,200),RAD(2,200)
C
C
C X1,X2,Y1,Y2 - ARRAYS FOR ENDPOINTS OF IMAGE LINES
C U1,U2,V1,V2 - ARRAYS FOR ENDPOINTS OF MAP LINES
C THET - ANGLES FOR IMAGE AND MAP
C RAD - RAD FOR IMAGE AND MAP
C NIMAGE - # OF LINES IN IMAGE (# OF ROWS IN MATRIX)
C NMAP - # OF LINES IN MAP (# OF COLUMNS IN MATRIX)
C IXCENT,IYCENT - X AND Y OF IMAGE CENTER
C MXCENT,MYCENT - X AND Y OF MAP CENTER
C RSIZE - # OF ROWS IN MATCHS
C CSIZE - # OF COLUMNS IN MATCHS
C
C NAMED COMMON REGIS IS USED TO HOLD INFO
C
C     COMMON /REGIS/ X1,X2,Y1,Y2,U1,U2,V1,V2,THET,RAD,
C             * NIMAGE,NMAP,IXCENT,IYCENT,MXCENT,MYCENT,RSIZE,CSIZE
C
C*** END OF MACRO REGIST, H.P. VERSION 9FEB79*****
C
C
C*****PROC BUCKET H.P. VERSION 9FEB79*****
C
C     DEFINES CLUSTER MATRICES, CLUSTERING IN 3-D (THETA,X,Y)
C     BKTCNT - CONTAINS COUNT OF # OF HITS IN EACH BUCKET
C     RKTOFF - CONTAINS COUNT OF # OF HITS IN OFFSET MATRIX
C
C     INTEGER BKTCNT(10,10,10),RKTOFF(10,10,10)
C     INTEGER NZEROT(10),NUMTHE,NUMX,NUMY,NZEROO(10)
C     COMMON /CLSTR/BKTCNT,RKTOFF,NZEROT,NZEROO,NUMTHE,NUMX,NUMY
C
C*** END OF PROC BUCKET ***
C
```

```
C
C*****PRTFLG MACRO H.P. VERSION 9FEB79*****
C
C      INTEGER PRTINP,PRTMWM,PRTBUK,PRTSBK
C
C      PRTINP - PRINT INPUT FLAG (RECTANGULAR AND POLAR)
C      PRTMWM - PRINT MATCH WEIGHT MATRIX FLAG
C      PRTBUK - PRINT 3-D CLUSTERING MATRIX FLAG (BUCKETS)
C      PRTSBK - PRINT SMOOTHED CLUSTERING MATRIX
C
C      NAMED COMMON PRTFL IS USED TO HOLD FLAGS
C
C      COMMON /PRTFL/ PRTINP,PRTMWM,PRTBUK,PRTSBK
C
C*** END OF PROC PRTFLG
C
C
C*****PROC CHSDF8 H.P. VERSION 9FEB79*****
C
C      INTEGER DX(8),DY(8)
C
C      DX,DY - CHANGES IN X AND Y FOR CHAIN CODES
C      THE NAMED COMMON DELTS HOLDS THESE VALUES AND IS INITIALIZED IN THE
C      BLOCK DATA PROGRAM
C
C      COMMON /DELTS/ DX,DY
C
C*** END OF PROC CHSDF8
C
```

## D.2 Possible commands

The command needed to get things started is the INPUT command.

The format is as follows (starting in column 1):

```
INPUT  <ANGTOL>, <STOL>, <DTOL>, <NUMLEV>, <PRTBUK>, <PRTOFF>, <PRTMWM>,
       <PWIDTH>, <LENCHK>, <THRESH>
```

Where the input fields have the following significance:

- ANGTOL - angular tolerance (in degrees) between model edge and a rotated image edge. Used in the routine ALMTCH.
- STOL - segment tolerance for generating the line segments in EDGMAP.
- DTOL - distance tolerance used in computing strength of match between a model edge and a transformed image edge. Calculation using DTOL is in routine EMATCH.
- NUMLEV - number of clustering levels desired. Maximum currently possible is 5. Used as a controlling parameter in REGDR.
- PRTBUK - flag for printing original clustering matrix. If value is 1, matrix is printed at each level.
- PRTOFF - flag for printing offset clustering matrix. If value is 1, matrix is printed at each level.
- PRTMWM - flag for printing match weight matrix.
- PWIDTH - number of columns available on output line. Can be either 72 or 132.
- LENCHK - if set to 1, only edges whose lengths are approximately the same will be compared.
- THRESH - minimum number of line segments that must pass through a bucket, at the highest clustering level, in order for that bucket to be considered as a peak.

All above inputs are integer.

The command to initiate the 3-dimensional clustering that does the bulk of the registration work, starts in column one and is CLUSTR.

A command that can be presented independently of the CLUSTR command, is the EMATCH command. This command allows the user to present a possible transformation and have the software evaluate the strength of the transformation given the edge information input as a result of the INPUT command processing. This command can be given before, after or in place of the CLUSTR command. The format is:

EMATCH <THETAI>,<XS>,<YS>,<DTOL>

where

THETAI - rotational angle, in degrees, of the transformation to be evaluated. (integer)

XS - X-shift of transformation to be evaluated (real)

YS - Y-shift of transformation to be evaluated (real)

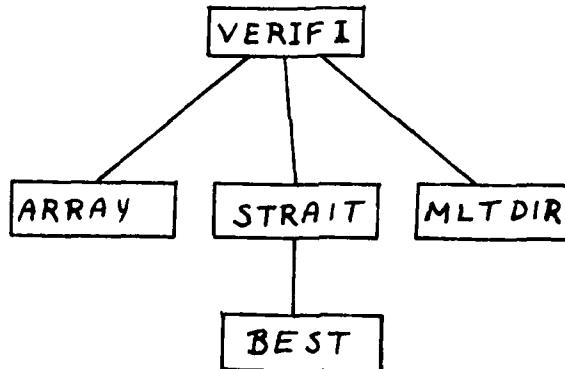
DTOL - distance tolerance to be used in computing strength of matches for this transformation. (integer)

The command to terminate processing is simply FINISH.

## Appendix E Verification Software

The software to do the verification of lineal features has been set up as an independent load module. The main program of this system is the program VERIFI. The software has the ability to verify both straight and curved lines. For curved lines, the input can either be presented one point at a time or the information can be stored in a data file which will be processed by the software.

The program structure is:



The verification program, given an inverse transformation, will check the image for the presence of known map features.

Routine Name	Arguments	Function
VERIFI	main program	Receives user input which defines the inverse transformation and the features to be verified. The user is prompted for the needed information. (An example execution is presented below) For each point on the hypothesized feature, a profile is created and the gradients along the profile are checked to see if the feature truly exists at this point. Execution can continue for several edges for the given transformation.
ARRAY	(DELTAX, DELTAY, SYVXLO, SYVYLO)	Given the change to be made in the X-direction (DELTAX) and in the Y-direction (DELTAY), the stage is moved the desired amount and the coordinates of the point where the stage is focused is returned in SYVXLO and SYVYLO.
STRAIT	(AX,AY,BX,BY,NNEIGB, STORAG,N,NLINKS, IFLAG)	Given the tail (AX,AY), the head (BX,BY) and the number of directions to use (4 or 8), the links of a straight line between the tail and head are placed into the array STORAG. If the number of links (NLINKS) is greater than the value of N, IFLAG is set to indicate an error. In this case, STRAIT is used to create the profiles.
BEST	(FX,FY,TX,TY,NNEIGB, L1,L2,N1,N2)	Calculates the straightest way to get from (FX,FY) to (TX,TY) depending on number of directions available for use (4 or 8)..
MLTDIR	(X,Y,RES,MAG,DIR, IERR)	See Appendix B for description of function.

The following are examples of 3 executions of the verification software. Computer prompts are identified by a CP in the left margin and user responses by a UR in the left margin.

Example 1: Curved line with point information stored in disk file.

```
UR      RUN VERII,6,6
CR      POSITION STAGE TO INITIAL POINT (BY HAND) AND INPUT COORDINATES
CR      FORMAT IS XPT,YPT (INT/FREE)
UR      10000,5000,
CR      GIVE VALUE FOR DTOL (INT/FREE)
UR      100,
CR      GIVE GRADIENT THRESHOLD VALUE (INT/FREE)
UR      4,
CR      GIVE DEBUG FLAG (VALUE BETW 0 & 4)
UR      3
CR      GIVE REVTHE,REVXS,REVYS,IXCENT,IYCENT,MXCENT,MYCENT
UR      342,-2433.,1701.,8000,8000,11000,11000,
CR      DTOL,REVTHE,REVXS,REVYX = 100  342  -2433.  1701.
CR      IXCENT,IYCENT,MXCENT,MYCENT = 8000  8000  11000  11000
CR      GIVE EDGE #
UR      1
CR      VERIFYING EDGE # 1
CR      IS THIS A STRAIGHT LINE?  INPUT 1 FOR YES, 0 FOR NO
UR      0
CR      IS INPUT IN A FILE?  (1 FOR YES, 0 FOR NO)
UR      1
CR      GIVE FILE NAME OF INPUT FILE - (3A2)
UR      CTDRUO
.
.
```

When processing of edge is completed and statistics have been printed, user will again be prompted for edge number. If there are no more edges to be processed input a -1. User will then be prompted for another reverse transformation, if none to be given, again input a -1.

Example 2: Curved line with point information given by user.

First 17 lines are the same as example 1.

CR IS THIS A STRAIGHT LINE? INPUT 1 for YES, 0 FOR NO  
UR 0  
CR IS INPUT IN A FILE? (1 FOR YES, 0 FOR NO)  
UR 0  
CR GIVE X,Y AND GRAD ANGLE VALUES (-1,-1,-1, FOR ENDING)  
UR 11032,10046,154,  
CR . give one point at a time  
UR .  
CR GIVE X,Y AND GRAD ANGLE VALUES (-1,-1,-1, FOR ENDING)  
UR -1,-1,-1

Stopping same as in example 1.

Example 3: Straight line with equally spaced sample points.

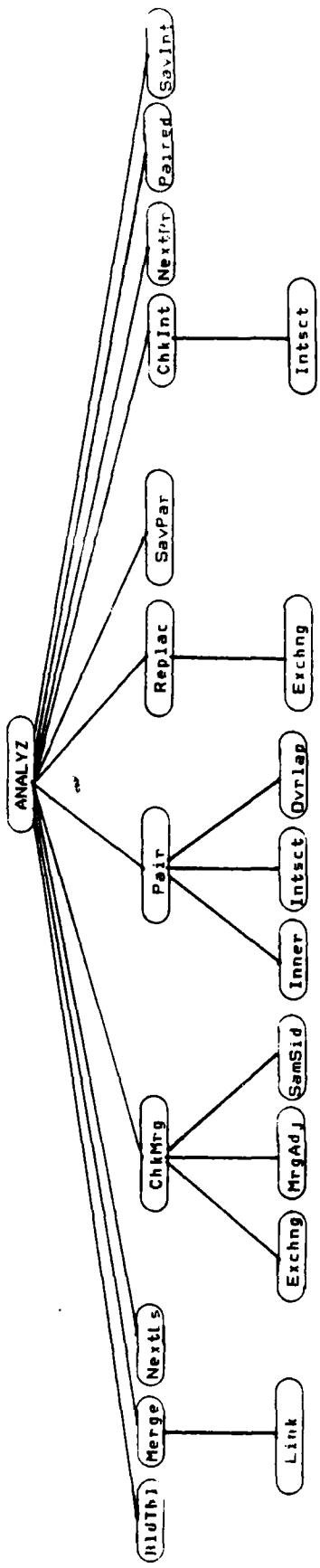
First 17 lines are the same as example 1.

CR IS THIS A STRAIGHT LINE? INPUT 1 FOR YES, 0 FOR NO  
UR 1  
CR GIVE XTAIL,YTAIL (INT/FREE)  
UR 4996,7282,  
CR GIVE DELTAX,DELTAY,GRAD,ANG AND NUMPTS (REAL,REAL,INT,INT)  
UR 55.65,113.32,334,41,

Stopping is accomplished in the same manner as in example 1.

Appendix F - This appendix contains documentation about the programs used to detect and classify intersections as discussed in Section 2.7.

*Structure of the Merging, Pairing, and Intersection System*



*Miscellaneous routines:*

Gravitp  
Dist  
OrtDis  
OrtInt

*Output routines:*  
PrtInt  
PrtList  
PrtTab  
PrtInt

These routines use Fortran IV and run on a UNIVAC 1100/40 under the EXEC 8 operating system.

```

1 FTL&ETL(1).ANALYZ(0)
2
3 C PROGRAM ANALYZ PROCESSES A LIST OF LINE SEGMENTS THAT HAVE BEEN
4 C EXTRACTED FROM AN IMAGE. PROCESSING CONSISTS OF THREE PARTS:
5 C Merging, Pairing, and Intersection. In merging, each line segment
6 C is compared to all others in an attempt to consolidate similar
7 C line segments into single line segments. The program then tries
8 C to pair all parallel regions of alternating light and dark shades.
9 C The results of merging and pairing are used in intersection to
10 C find line segments and pairs of line segments that are positioned
11 C in the forms of X's, T's, and L's.
12 C
13 C INPUT: LIST OF LINE SEGMENTS.
14 C
15 C OUTPUT FROM MERGING: NEW LIST OF LINE SEGMENTS CONTAINING
16 C THE NEW MERGED LINE SEGMENTS AND ANY
17 C OLD LINE SEGMENTS THAT COULD NOT BE
18 C MERGED.
19 C
20 C OUTPUT FROM PAIRING: NEW LIST OF LINE SEGMENTS POSSIBLY CON-
21 C TAING MERGED LINE SEGMENTS (SEE PAIRING
22 C PART OF PROGRAM), AND A LIST OF PAIRED
23 C LINE SEGMENTS.
24 C
25 C OUTPUT FROM INTERSECTION: A LIST OF THE INTERSECTIONS FOUND AND
26 C THEIR TYPES.
27 C
28 C GRAPHICS OUTPUT: IF THE ANALYZ PROGRAM IS INVOKED BY THE COMMAND
29 C 'EXET,O FILE,ANALYZ', THEN ANY TIME THE SEGMENT
30 C LIST IS OUTPUT, A GRAPHICS FILE WILL BE WRITTEN TO
31 C UNIT 10 (WHICH MUST BE ASSIGNED & ALIASED BEFORE).
32 C EACH LINE SEGMENT LIST ON FILE 10 IS FORMATTED FOR
33 C USE WITH THE 'GRAPH' COMMAND ON THE VAX UNIX OS.
34 C
35 C COMMON VARIABLES: /DRADTB/: DELTA-RADIUS TABLE.
36 C DRAD(N) - DELTA-RADIUS VALUE FOR THE
37 C CURRENT LINE SEGMENT AND SEGMENT N.
38 C
39 C /LIST/: INTERSECTION LIST.
40 C ICOUNT - NUMBER OF ENTRIES IN THE LIST.
41 C INTLIN(N,1..4) - INDICES OF UP TO 4 LINE
42 C SEGMENTS THAT FORM AN INTERSECTION.
43 C XINSET, YINSET(N) - COORDINATES FOR THE
44 C POINT OF INTERSECTION.
45 C
46 C /PTABLE/: PAIR TABLE.
47 C PCOUNT - NUMBER OF PAIRS IN TABLE.
48 C PRTABL(N,1) & PRTABL(N,2) - INDICES OF TWO
49 C LINE SEGMENTS THAT HAVE BEEN PAIRED.
50 C PRTABL(N,3) - CODE INDICATING THE TYPE OF
51 C PAIRING.
52 C
53 C /SLIST/: SEGMENT LIST.
54 C SCOUNT - # OF LINE SEGMENTS IN THE LIST.
55 C AX, AY, BX, BY(N) - ENDPOINT COORDINATES
56 C

```

57        C  
 58        C  
 59        C  
 60        C  
 61        C  
 62        C  
 63        C  
 64        C  
 65        C  
 66        C  
 67        C  
 68        C  
 69        C  
 70        C  
 71        C  
 72        C  
 73        C  
 74        C  
 75        C  
 76        C  
 77        C  
 78        C  
 79        C  
 80        C  
 81        C  
 82        C  
 83        C  
 84        C  
 85        C  
 86        C  
 87        C  
 88        C  
 89        C  
 90        C  
 91        C  
 92        C  
 93        C  
 94        C  
 95        C  
 96        C  
 97        C  
 98        C  
 99        C  
 100      C  
 101      C  
 102      C  
 103      C  
 104      C  
 105      C  
 106      C  
 107      C  
 108      C  
 109      C  
 110      C  
 111      C  
 112      C  
 113      C

\*\*\*\*\* ANALYZE \*\*\*\*\*  
 FOR THE LINE SEGMENTS:  
 NUMBS(N) - NUMBER OF LINES THAT SEGMENT N  
 HAS BEEN PAIRED WITH.  
 FLAG(N) - IF N HAS BEEN MERGED WITH M (N<N),  
 THEN FLAG(N)=M (POINTS TO MERGED LINE  
 COORDINATES). IF N(N) HAS NOT BEEN  
 MERGED, THEN FLAG(N)=0.

MAIN SUBROUTINES:

MERGE - ATTEMPTS TO MERGE TWO LINE SEGMENTS, A->B & C->D.  
 IF THE ATTEMPT IS SUCCESSFUL, THEN THE COORDINATES  
 OF THE MERGED LINE ARE RETURNED THROUGH THE COORD-  
 INATE ARGUMENTS FOR THE LINE SEGMENT A->B, AND CODE  
 INDICATES THE TYPE OF MERGING:  
 CODE = 0 : NO MERGE.  
 = 1 : 1-TH SEGMENT DOMINATES J-TH SEGMENT WITH  
 LENGTH(I) > LENGTH(J).  
 = 2 : AS IN 1, BUT LENGTH(I) > LENGTH(J).  
 = 3 : J-DOMINATION, LENGTH(J) > LENGTH(I).  
 = 4 : AS IN 3, BUT LENGTH(J) > LENGTH(I).  
 = 5 : FORWARD (I-J) LINKING, I.E., LINK THE HEAD  
 OF A->B (I) TO THE TAIL OF C->D (J).  
 = 6 : BACKWARD (J-I) LINKING, I.E., LINK THE  
 HEAD OF C->D (J) TO THE TAIL OF A->B (I).

PAIR - TWO LINE SEGMENTS ARE TESTED TO SEE IF THEY DELINATE  
 PARALLEL REGIONS OF ALTERNATING LIGHT AND DARK SHADES.  
 IF PAIRING IS POSSIBLE, THEN CODE IS SET TO 1 IF THE  
 TWO LINE SEGMENTS REPRESENT A LIGHT STRIPE ON A DARK  
 BACKGROUND, OR TO 2 IF A DARK STRIPE ON A LIGHT BACK-  
 GROUND IS REPRESENTED.

CONSTANTS AND PARAMETERS:  
 MAXENT/100/ : ALL ROUTINES - THE MAXIMUM # OF ENTRIES FOR  
 ARRAYS IN THE COMMON BLOCKS.  
 NULL/-1.0/ : BLDTBL, NEXTLS - INDICATES NULL ENTRY IN THE  
 DRAID TABLE.  
 QLPTH/0.50/ : PAIR - FRACTION OF OVERLAP THRESHOLD.  
 PI/3.141592654/ : CHKINT, INTST, MERGE, PAIR  
 REP8LN/7.0/ : MERGE - TOLERANCES IN TESTING ENDPOINT RELA-  
 TIONS BETWEEN VECTORS.  
 SEPSLN/7.0/ : MERGE - MAXIMUM DELTA-THETA ALLOWED  
 TEPSLN/0.2618/ :  
 TEPSLN/0.175/ : CHKINT, PAIR  
 TOLRNC/7.0/ : TOL : CHMRC - DISTANCE TOLERANCE.  
 UPREP8/25.0/ : PAIR - MAXIMUM DELTA-RADIUS ALLOWED.

COMPILE, MAPPING, AND EXECUTION:  
 EACH ROUTINE IS COMPILED USING 'FOR FILE NAME'.

```

114 C MAPPING USES THE FOLLOWING - EXECUTABLE ELEMENT.
115 C 'GMAP, I ,FILE, ANALYZ
116 C IN FILE, ANALYZ
117 C IN FILE, MERGE, PAIR
118 C IN FILE, BLDTBL, NEXTLS, PRTLIS, PRTAB
119 C IN FILE, CNTRP, DIST, INTSCT, ORTDIS, ORTINT
120 C IN FILE, LIMK-6VEND
121 C IN FILE, CHNRG, EXCHNG, HRCADJ, REPLAC, SAMSID, SAVPAR
122 C IN FILE, CHKINT, NEXTPR, PAIRED, SAVINT
123 C NOT TPFs.
124 C
125 C
126 C EXECUTION (SEE THE NOTE ABOVE ON GRAPHICS OUTPUT) -
127 C 'EXQT FILE, ANALYZ
128 C X1 Y1 X2 Y2 : FREE FORMAT INPUT OF THE ENDPOINT
129 C : COORDINATES FOR THE LINE SEGMENTS.
130 C
131 C
132 C
133 C
134 C
135 C
136 INTEGER CODE, I, J, LWRBND, NPRSJ, NPREJ, NXPRI, NXPRI, NUMBER
137 INTEGER SEGI, SEQJP, SEQJ, SEQJP, UPRBND, WHERE, XINTER, YINTER
138 LOCAL CHANGE, GRAPH, PROCES
139 LOCAL OPT, PAIRED
140 LOCAL OPT, PAIRED
141 INTEGER ICOUNT, INTLIN, MAXENT/100, TYPE
142 REAL XINSET, YINSET
143 COMMON /ILIST/ ICOUNT, INTLIN(100,4), XINSET(100),
144 COMMON /ILIST/ ICOUNT, INTLIN(100,4), XINSET(100),
145 COMMON /ILIST/ ICOUNT, INTLIN(100,4), XINSET(100),
146 INTEGER PCOUNT, PRTABL
147 COMMON /PTABLE/ PCOUNT, PRTABL(100,3)
148
149
150 INTEGER FLAG, NUMPRS, SCOUNT
151 REAL AX, AY, BX, BY
152 COMMON /SLIST/ SCOUNT, AX(100), AY(100), BX(100),
153 COMMON /SLIST/ SCOUNT, AX(100), AY(100), BX(100),
154 COMMON /SLIST/ SCOUNT, AX(100), AY(100), BX(100),
155
156
157 C. . . GET THE GRAPH OPTION FROM THE 'EXQT' JCL CARD.
158
159
160 C. . . SCOUNT = FALSE.
161 IF ( OPT ('0') ) GRAPH = TRUE.
162
163 C. . . INPUT THE LINE SEGMENTS INTO THE SEGMENT LIST, SLIST.
164
165 C. . . SCOUNT = 0
166 SCOUNT = 0
167 SCOUNT = 0
168 SCOUNT = 0
169 SCOUNT = SCOUNT + 1
170 READ (5,400,END=20) AX(SEGI), AY(SEGI), BX(SEGI), BY(SEGI)

```

```

171 400 READ (5,400,END=20) AX(SEG1), AY(SEG1), BX(SEG1), BY(SEG1)
172 FORMAT ()
173 NUMPRS(SEG1) = 0
174 FLAG(SEG1) = 0
175 SCOUNT = SCOUNT + 1
176 IF (SCOUNT .LT. MAXENT) GOTO 10
177
178 20 WRITE (6,500)
179 500 FORMAT ('0', 'INPUT LIST OF LINE SEGMENTS: ')
180 CALL PRTLIS (GRAPH)
181
182
183 C **** MERGING ****
184 C
185 C THE MERGING PORTION OF THE PROGRAM TRIES TO MERGE EACH LINE
186 C SEGMENT IN THE LIST WITH ALL THE OTHER LINE SEGMENTS. THE
187 C STRUCTURE OF THIS PART IS A DO-LOOP THAT STEPS THROUGH THE
188 C SEGMENT LIST. EACH PASS THROUGH THE LOOP DOES THE FOLLOWING:
189 C
190 C 1) BUILD THE 'DELTA-RADIUS' TABLE, CONTAINING THE ABSOLUTE
191 C VALUES OF THE DIFFERENCES BETWEEN THE POLAR RADII OF THE
192 C CURRENT LINE SEGMENT AND ALL OTHER LINE SEGMENTS.
193 C 2) IN ORDER OF INCREASING DRAD'S, ATTEMPT TO MERGE THE CURRENT
194 C LINE SEGMENT WITH ALL THE OTHER SEGMENTS.
195 C 3) IF 2 LINE SEGMENTS ARE MERGED, THE CURRENT LINE WILL THEN
196 C HAVE NEW COORDINATES. THEREFORE, GO TO STEP 1, REBUILD
197 C THE DRAD TABLE USING THE CURRENT LINE SEGMENT (NOW MERGED),
198 C AND REPEAT THE PASS.
199 C
200 C WHEN TWO LINE SEGMENTS ARE MERGED, THE DELETED LINE SEGMENT IS
201 C FLAGGED TO POINT TO THE MERGED LINE.
202
203
204 205 IF (SCOUNT .LE. 1) GOTO 45      * ENOUGH FOR MERGING?
205 UPBND = SCOUNT - 1
206 DO 40 SEG1 = 1, UPBND
207 40 IF (FLAG(SEG1) .GT. 0) GOTO 40      * ALREADY MERGED?
208
209 210 CALL BLDTBL (SEG1)      * BUILD THE 'DRAD' TABLE.
210
211 212 SEGJ = NEXTLS (SEG1)      * GET LINESEG WITH SMALLEST DRAD.
213 IF (SEGJ .LE. 0) GOTO 40      * EMPTY DRAD TABLE?
214 IF (FLAG(SEGJ) .GT. 0) GOTO 35      * SAFETY CHECK.
215
216 CALL MERGE (AX(SEG1), AY(SEG1), BX(SEG1), BY(SEG1),
217 + AX(SEGJ), AY(SEGJ), BX(SEGJ), BY(SEGJ), CODE)
218
219 220 IF (CODE .LE. 0) GOTO 35      * NO MERGE?
220 FLAG(SEGJ) = SEG1      * FLAG DELETED LINE SEGMENT.
221 GOTO 30      * LOOP TO REBUILD 'DRAD' TABLE.
222
223 224 CONTINUE      * END DO-LOOP.
224
225
226 227 45 WRITE (6,510)

```

228 510 FORMAT (////1X, 'MERGED LINE SEGMENTS LIST: ')

229 CALL PRTLIB (GRAPH)

230

231

232

233

234 \*\*\*\* PAIRING \*\*\*\*\*

235 C FAIRING (IN/MERGING) TAKES A SET OF LINE SEGMENTS (VECTORS) AND  
236 C TRIES TO PAIR ALL COMBINATIONS OF APPROXIMATELY PARALLEL SEGMENTS.  
237 C A LIST IS MADE OF ALL THE PAIRS FOUND. THIS LIST IS THEN PROCESSED  
238 C TO FIND IF THE POSSIBILITY OF MERGING EXISTS FOR ANY INSTANCES OF  
239 C THE FOLLOWING SITUATION:

240 C LINE SEGMENT L1

241 C

242 C

243 C

244 C

245 C

246 C

247 C

248 C

249 C

250 C

251 C

252 C

253 C

254 C

255 C

256 C

257 C

258 C

259 C

260 C

261 C

262 C

263 C

264 C

265 C

266 C

267 C

268 C

269 C

270 C

271 C

272 C

273 C

274 C

275 C

276 C

277 C

278 C

279 C

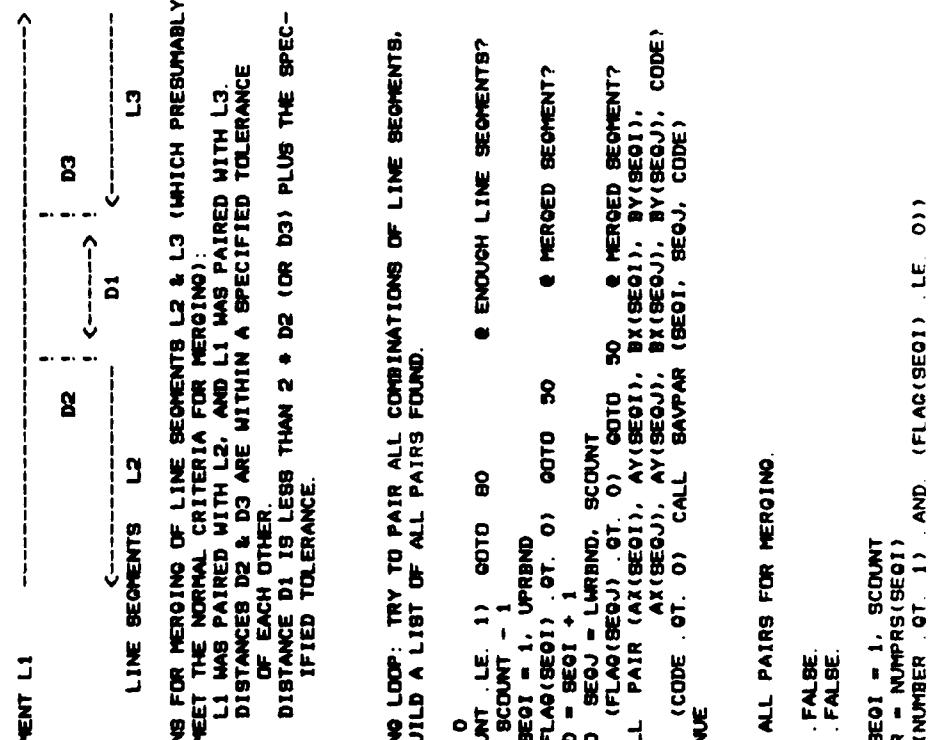
280 C

281 C

282 C

283 C

284 C



PAIRING LOOP: TRY TO PAIR ALL COMBINATIONS OF LINE SEGMENTS,  
AND BUILD A LIST OF ALL PAIRS FOUND.

```
PCOUNT = 0
IF (SCOUNT .LE. 1) GOTO 80      * ENOUGH LINE SEGMENTS?
UPRND = SCOUNT - 1
DO 50 BEG1 = 1, UPRND
  IF (FLAG(SEQ1) .GT. 0) GOTO 30  * MERGED SEGMENT?
  LWRND = BEG1 + 1
  DO 30 SEQ2 = LWRND, SCOUNT
    IF (FLAG(SEQ2) .GT. 0) GOTO 30  * MERGED SEGMENT?
    CALL PAIR (AX(SEQ1), AY(SEQ1), BX(SEQ1), BY(SEQ1), CODE)
    + IF (CODE .GT. 0) CALL BAVPAR (SEQ1, SEQ2, CODE)
  CONTINUE
30
```

C CHECK ALL PAIRS FOR MERGING.

```
PROCES = FALSE.
CHANGE = FALSE.
DO 60 SEQ1 = 1, SCOUNT
  NUMBER = NUMPS(SEQ1)
  IF ((NUMBER .GT. 1) .AND. (FLAG(SEQ1) .LE. 0))
```

```

285 60      + CALL CHKPRO (1801, NUMBER, CHANGE)
286      CONTINUE
287
288 PROCES = PROCES OR CHANGE      * CLEAN UP PAIR TABLE.
289 IF (CHANGE) CALL REPLAC      * REPEAT UNTIL NO CHANGE.
290 IF (CHANGE) GOTO 55
291
292
293 C   IF ANY LINES WERE MERGED, PRINT THE REVISED LINE SEGMENT LIST.
294 IF (6, NOT, PROCES) GOTO 70      * ANY LINES MERGED?
295 WRITE (6, 520)
296 520 FORMAT ('////1X, 'MERGING DURING PAIRING - REVISED LINE SEGMENT '
297      + 'LIST: ')
298      CALL PRTLIS (GRAPH)
299
300 C   PRINT THE PAIR TABLE.
301
302 70 WRITE (6, 530)
303 530 FORMAT ('////1X, 'PAIRED LINE SEGMENTS: ')
304      CALL PRITAB
305      GOTO 90
306
307
308
309
310 80 WRITE (6, 540)
311 540 FORMAT ('//1X, 'INsUFFICIENT NUMBER OF LINE SEGMENTS FOR PAIRING. ')
312
313
314
315
316
317
318 C...
319 C   **** INTERSECTION ****
320 C   IN THIS PART OF THE PROGRAM, THE POSITIONING AND PAIRING OF
321 C   LINE SEGMENTS ARE EXAMINED FOR THE EXISTENCE OF X-, T-, AND L-
322 C   TYPE INTERSECTIONS. THE DIFFERENT TYPES OF INTERSECTIONS RECO-
323 C   GNIZED AND THEIR CODES ARE:
324 C
325 C   CODES: 1 - X1: 2 LINE SEGMENTS CROSSING.
326 C           2 - X2: 2 PAIRED LINE SEGMENTS CROSSED BY A THIRD.
327 C           3 - X3: 2 PAIRS OF LINE SEGMENTS CROSSING EACH OTHER.
328 C
329 C   4 - T1: 2 SEGMENTS IN A T-SHAPE.
330 C           5 - T3: 2 PAIRED SEGMENTS AND A THIRD IN A T-SHAPE.
331 C           6 - T2: 2 PAIRS OF LINE SEGMENTS IN A T-SHAPE.
332 C
333 C   7 - L1: 2 SEGMENTS IN AN L-SHAPE.
334 C           8 - L3: 2 PAIRED SEGMENTS AND A THIRD IN AN L-SHAPE.
335 C           9 - L2: 2 PAIRS OF LINE SEGMENTS IN AN L-SHAPE.
336 C
337 C
338 C   THE SEARCH FOR INSTANCES OF INTERSECTION PROCEEDS AS FOLLOWS:
339 C
340 C   FOR EACH LINE SEGMENT I IN THE SEGMENT TABLE
341 C   FOR I AND EACH LINE SEGMENT PAIRED WITH I

```

```

342 C FOR J AND EACH LINE SEGMENT PAIRED WITH J (BUT NOT
343 C WITH I OR ITS PAIRS)
344 C CHECK FOR THE DIFFERENT CASES OF INTERSECTION.
345 C
346 C IF A CASE OF INTERSECTION IS FOUND, THE LINE SEGMENTS AND THE CODE
347 C ARE SAVED IN THE INTERSECTION TABLE. THE 3 INNER LOOPS ARE MORE
348 C RESTRICTED IN THE ACTUAL PROGRAM IN ORDER TO PREVENT DUPLICATE DE-
349 C TECTIONS OF INTERSECTIONS.
350
351
352 90 IF (SCOUNT .LE. 1) GOTO 180 ④ ENOUGH LINES?
353
354 1COUNT = 0
355
356 UPRBND = SCOUNT - 1
357 DO 140 SEGI = 1, UPRBND
358 IF (FLAG(SEGI) .GT. 0) GOTO 140 ④ MERGED LINE?
359
360 NPRS1 = NUMPRS(SEGI)
361 IF (NPRS1 .LE. 0) NPRS1 = 1 ④ CHECK SEGI ALWAYS.
362 NXPRI = 1 ④ POINTER INTO PAIR TABLE.
363
364 DO 130 I = 1, NPRS1
365
366 CALL NEXTPR (SEGI, SEGIP, NXPRI)
367 NXPRI = NXPRI + 1
368 IF ((FLAG(SEGIP) .GT. 0) .OR. (SEGI .GT. SEGIP)) GOTO 130
369
370 LMRBND = SEGI + 1
371 DO 120 SEQJ = LMRBND, SCOUNT
372 IF (FLAG(SEQJ) .GT. 0) GOTO 120
373 IF (PAIRED (SEGI, SEQJ)) GOTO 120
374 IF (PAIRED (SEGIP, SEQJ)) GOTO 120
375
376 NPRSJ = NUMPRS(SEQJ)
377 IF (NPRSJ .LE. 0) NPRSJ = 1 ④ CHECK SEQJ ALWAYS.
378 NXPRI = 1 ④ POINTER INTO PAIR TABLE.
379 NXPRI = 1
380
381 DO 110 J = 1, NPRSJ
382
383 CALL NEXTPR (SEQJ, SECJP, NXPRI)
384 NXPRI = NXPRI + 1
385 IF ((FLAG(SECJP) .GT. 0) .OR. (SEQJ .GT. SECJP))
386 GOTO 110
387 IF (PAIR (SEGI, SECJP)) GOTO 110
388 IF (PAIRED (SEGIP, SECJP)) GOTO 110
389
390 CALL CHKINT (SEGI, SECJ, XINTER, YINTER, WHERE)
391 IF (WHERE .GT. 0) GOTO 100
392
393 IF (SECIP .NE. SEGI)
394 CALL CHKINT (SECIP, SECJ, XINTER, YINTER, WHERE)
395 IF (WHERE .GT. 0) GOTO 100
396
397 IF (SECJP .NE. SECJ)
398 CALL CHKINT (SECJ, SECJP, XINTER, YINTER, WHERE)

```

```

***** ANALYZ *****

399      +          CALL CHKINT (SEQ1, SEQJP, XINTER, YINTER, WHERE)
400      +          IF (WHERE .GT. 0) GOTO 100
401
402      +          IF ((SEQ1P .NE. SEQ1) .AND. (SEQJP .NE. SEQJ),
403          +          CALL CHKINT (SEQ1P, SEQJP, XINTER, YINTER, WHERE)
404          +          IF (WHERE .LE. 0) GOTO 110
405
406      C...  INTERSECTION FOUND. SET THE CODE FOR INTERSECTION AND BAVE THE
407      C...  INFORMATION IN THE INTERSECTION LIST.
408      100      CODE = 3 * WHERE - 2
409          +          IF ((SEQ1P .NE. SEQ1) .OR. (SEQJP .NE. SEQJ))
410          +          CODE = CODE + 1
411          +          IF ((SEQ1P .NE. SEQ1) .AND. (SEQJP .NE. SEQJ))
412          +          CODE = CODE + 1
413          +          CALL SAVINT (SEQ1, SEQJP, SEQ1P, SEQJP,
414          +          XINTER, YINTER, CODE)
415
416      C...  CONTINUE
417      110      CONTINUE
418      120      CONTINUE
419      130      CONTINUE
420      140      CONTINUE
421
422
423      C...  PRINT THE INTERSECTION LIST.
424      424      WRITE (6, 550)
425      425      FORMAT (//1X, 'INTERSECTIONS: ')
426      550      CALL PRTINT
427      570      GOTO 190
428
429
430
431      180      WRITE (6, 560)
432      560      FORMAT (//1X, 'INSUFFICIENT NUMBER OF LINE SEGMENTS FOR ',
433          +          'INTERSECTION. ')
434
435
436
437      437      C...  *** END OF PROGRAM ***
438
439
440      440      190      WRITE (6, 570)
441      570      FORMAT (//1X, 'END OF PROGRAM, PAIRING, AND INTERSECTION. ')
442
443
444
445
446

```

